CHALMERS
UNIVERSITY OF TECHNOLOGY

Generalization of Monocular Semantic BEV Prediction in Low Annotated Regimes for Unstructured Off-Road Environments

Master's thesis in Complex Adaptive Systems

Max Magnusson
Mattias Lind

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Generalization of Monocular Semantic BEV Prediction in Low Annotated Regimes for Unstructured Off-Road Environments

MAX MAGNUSSON
MATTIAS LIND

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Generalization of Monocular Semantic BEV Prediction in Low Annotated Regimes for Unstructured Off-Road Environments

Max Magnusson, Mattias Lind

Supervisor: Dag Bergsjö, Eira Systems
Examiner: Lars Hammarstrand, Chalmers, Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A visualization of semantic mask for monocular image in off-road terrain.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Generalization of Monocular Semantic BEV Prediction in Low Annotated Regimes for Unstructured Off-Road Environments

Max Magnusson, Mattias Lind
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Autonomous navigation in unstructured off-road environments presents distinct challenges from urban driving. Terrain boundaries are ambiguous, annotated datasets are scarce, and in military contexts active sensors such as LiDAR or radar carry the risk of revealing a vehicle's position. This thesis addresses the problem with a weakly-supervised perception pipeline that relies on passive camera input at inference time, using LiDAR only offline to provide geometric supervision during training. The Vision Foundation Model CAT-Seg is used to generate semantic pseudo-labels across five collapsed traversability classes. These are projected into Bird's Eye View space using LiDAR-assisted depth integration from DepthAnythingV2 calibrated by LiDAR through RANSAC alignment, producing 100×100 BEV semantic grids as training targets without any manual annotation. Three BEV architectures, Lift Splat Shoot, FocusBEV and SimpleBEV, are trained under three supervision regimes: manually annotated ground truth (≈1,000 frames), VLM-derived pseudo-labels (≈18,000 frames), and a pseudo-label model fine-tuned on annotated data. The Great Outdoors dataset is used for in-distribution testing and RELLIS-3D for out-of-distribution testing. In distribution, pseudo-label trained models perform comparably to manually supervised ones despite lower label quality, with the large unlabeled volume compensating for pseudo-label noise. Out of distribution, pseudo-label training improves generalization across all three architectures, most clearly for LSS (0.141 to 0.214 mIoU) and FocusBEV (0.151 to 0.199 mIoU), inverting the in-distribution ranking where annotated training was equal or better. SimpleBEV consistently achieves the strongest out-of-distribution performance across training conditions in our experiments. These results suggest that VLM-produced pseudo-labeling can be a viable alternative to manual annotation for off-road BEV perception when generalization to new terrain is the primary concern.

Keywords: off-road autonomy, LiDAR, UGV, weak-supervision, Vision Foundation Models, Bird's Eye View, pseudo ground-truth, manual annotation.

Acknowledgements

We would like to thank Dag Bergsjö our supervisor for the opportunity to do our thesis at Eira, our examiner Lars Hammarstrand for valuable feedback. We would also like to thank our families for all the support during the time-period of this work and during the studies at Chalmers. Finally, we want to thank our friends made during our studies for making these years much more enjoyable.

Mattias Lind, Max Magnusson, Gothenburg, June 2026

Division of Labor

The authors in accordance with provided feedback, declare that the work load was evenly split.

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AI	Artificial Intelligence
BEV	Bird's Eye View
CLIP	Contrastive Language-Image Pre-training
FiLM	Feature-wise Linear Modulation
GT	Ground truth
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LSS	Lift Splat Shoot
mIoU	Mean Intersection over Union
UGV	Unmanned Ground Vehicle
VFM	Vision Foundation Model
VLM	Vision-Language Model
ViT	Vision Transformer
WCE	Weighted Cross-Entropy

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i: Index for pixels

j: Index for voxels

c: Index for semantic classes

d: Index for discrete depth bins

Sets

V: The predefined 3D metric voxel grid

$\mathcal{C}$: The set of semantic classes

$\mathcal{D}$: The set of discrete depth intervals used in forward projection

Parameters

K: Camera intrinsic matrix

$T_{\text{cam}\to\text{ego}}$: Extrinsic transformation matrix

f_x, f_y: Focal lengths in pixels

c_x, c_y: Principal point offsets

d_k: Dimension of key vectors

w_c: Class weight

ϵ: Smoothing constant

Variables

P_c: 3D point in camera frame

u, v: 2D pixel in image plane

Z_c: Depth value of a point along the optical axis

$\alpha(u, v)$: Predicted categorical depth distribution

$c(u, v)$: Feature vector from image

Q, K, V: Query, Key, and Value matrices

β: shifting parameter

γ: scaling parameter

$y, \hat{y}$: Ground truth label and predicted probability distribution

$\mathcal{L}$: The scalar value of a loss function

Contents

List of Figures

List of Tables

1

Introduction

The progress of autonomous driving systems in urban environments has been largely driven by massive, human-annotated datasets and their availability [1, 2, 3, 4]. These formatted datasets have been crucial part for the development of deep learning for on-road perception. This has also driven the development of autonomous systems in unstructured off-road terrain although the subfield suffers from serious data limitations and availability [5, 6, 7, 8]. Off-road environments show large visual variability across different terrains [9]. Meaning that relying only on manually annotated datasets is difficult to scale, highlighting the need for adaptable, automated labeling approaches [10]

In off-road autonomy, the distinction between drivable terrain and hazards becomes ambiguous. The transition from dirt to mud, or grass to bush, lacks the sharp boundaries found in the structured urban scene [11]. As a result of this, generating pixel-perfect semantic annotations for these environments is expensive, time consuming as well as largely subjective [10]. Because of the defense-oriented nature of many off-road systems, high-quality datasets are often restricted [12]. Since scaling perception models using strictly supervised learning on limited, manually labeled data is impractical [13], alternative self-supervised, weakly-supervised, or semi-supervised methods should be evaluated. However, recent papers have also explored the possible use of self-supervised autonomous driving methods at runtime, such as RoadRunner M&M [14].

This thesis investigates the use of pseudo-labeling as an alternative or an addition to human-annotated data in off-road autonomy. This work studies the use of Vision-Language Models to automatically generate semantic pseudo-labels. By combining the semantic masks created using the VLM together with LiDAR sweeps, we construct a "pseudo-ground truth" dataset entirely offline.

The focus of this research is to evaluate how viable pseudo-labeled data is when used to train downstream 2D-to-3D Bird's-Eye View perception models. By comparing models trained on machine-generated data against those trained on human-annotated data, this project aims to answer the following research questions:

- How does a BEV segmentation model trained using VLM-generated pseudo-labels compare to one trained on manual annotation in terms of mIoU and class IoU?
- To what degree do pseudo-labels from VLM capture unstructured off-road

boundaries compared to human ground truth?

- Can BEV models trained on semantic labels generalize to unseen out-of-distribution data, and to what extent?

1.1 Scope and Demarcation

This thesis investigates the use of pseudo-labeling as a way to remove or reduce the need for manually annotated data in off-road UGV perception. The study considers how data quality and volume impact the performance of multi-class semantic BEV prediction networks.

1.1.1 Contributions

While previous work has explored BEV perception on-road, BEV perception on unstructured terrain with limited annotations remains an open challenge. The contributions of this thesis are the following.

The development and review of a data preparation pipeline that uses zero-shot Vision-Language Models to generate semantic segmentation of images in unstructured terrain, reducing the need for manual annotation during training data generation.

A analysis of the noise, boundary errors, and class biases in VLM-generated pseudo-labels when compared to manually annotated ground truth across four semantic classes.

Evaluation of three BEV prediction architectures under three training methods. Only pseudo-labeled data, only annotated data, and a combination of the two with models pretrained on pseudo-labeled data and fine-tuned with annotated data.

Testing the performance of models trained on pseudo-labels and how they compare to manually supervised models on out-of-distribution data.

1.1.2 Delimitations

Because this thesis focuses on the reduction of the data limitation bottleneck in off-road terrain, the scope is bound to offline data generation and local training of models. As a result, broader autonomous system tasks such as global mapping, SLAM, and route planning are not addressed. While the models output local semantic maps, the downstream interpretation of these maps falls outside the scope of this work. Active sensors like LiDAR are strictly for offline use. LiDAR data is used to generate the ground truth targets. These targets are necessary for training and validation. They play no role in the runtime inference of the evaluated BEV models. The zero-shot Vision Foundation Models used for the auto-labeling of images are pre-trained. The underlying pre-training of these massive models is not explored and neither is the potential fine-tuning of them.

1.2 Thesis Outline

Chapter 1 introduces the problem of data scarcity in unstructured off-road environments and defines the research objectives, contributions, and scope.

Chapter 2 reviews related work concerning off-road datasets, the data scarcity problem, and research within off-road perception.

Chapter 3 establishes the theoretical basis of camera geometry, foundation models, and knowledge distillation.

Chapter 4 details the methodology of the pseudo-label generation pipeline and the specifications of the evaluated BEV architectures.

Chapter 5 presents the performance metrics of the different training methods of the downstream BEV models.

Chapter 6 discusses the meaning of these results regarding dataset scaling.

Chapter 7 concludes the thesis.

2

Related Work

2.1 Semantic Segmentation in Off-Road Terrain

A paper on semantic segmentation in off-road environment introduces the segmentation model OFFSEG [15]. They emphasizes on the added difficulty of off-road navigation where semantic and geometric understanding of the terrain is crucial. The difficulty of the problem is increased since quality datasets are highly limited. Many datasets also exhibit class imbalances which results in poor mIoU scores.

To overcome this, it was suggested that class labels in off-road datasets should be composed in to a few larger groups. This was due to the fact that certain class distinctions were considered unnecessary. A framework used for off-road driving does not necessarily gain any useful information by the differentiation between grass and hay. The result was combining classes into distinct groups. Sky, traversable, non-traversable and obstacles. This mitigated the class imbalance. Then, when a traversable area is found, the model uses K-Means clustering and a MobileNetV2 classifier to identify sub-classes like mud, gravel, and puddles.

2.2 Coarse Annotation

Recent research [16] has explored the use of coarse pixel annotation, as a low cost but useful alternative to fine annotation for urban scene semantic segmentation. Where coarse labels take roughly 7 minutes per image versus about 90 for fine labels, and yield over 10× more data for a fixed budget, but suffer from missing boundary information that makes them hard to train on directly. To use them, the authors propose a coarse to fine self-training framework that combines coarsely annotated real images with densely annotated synthetic data, using a boundary loss on the synthetic data to recover class boundaries, cross-domain copy-paste augmentation to bridge the real synthetic gap, and iterative pseudo-labeling to fill in the unlabeled boundary regions of the coarse data. On Cityscapes [1] and BDD100k [17] they reach mIoU competitive with fully fine-annotated training at roughly 1/5 and 1/8 of the annotation cost respectively, with the gains largest in low budget regimes and on rare classes, show that annotating many diverse coarse examples matters more than labeling every pixel precisely.

2.3 BEV perception

A common and efficient representation of image data in the field of autonomous driving is Bird's-Eye-View. BEV provides a top-down perspective useful for planning and perception tasks. The perspective is a key part when searching for the optimal path and assessing traversability of objects. Traditional methods for transforming perspective image features into a unified BEV space often relied on explicit depth estimation, which is known limitation.

BEVFormer [18] introduced a framework that bypasses these limitations using spatiotemporal transformers. Instead of projecting pixels based on depth, BEVFormer employs grid-shaped BEV queries that look up features from multi-camera images via spatial cross-attention. This allows the network to learn the 3D transformation adaptively. It incorporates temporal self-attention to recurrently fuse historical BEV features. This temporal information provides a resilient baseline for off-road navigation. This helps the model predict occluded objects in uncertain regions and increases the accuracy of velocity data when camera views are compromised.

2.4 Multi-layered BEV representations

TerrainNet [19] is a model and architecture for high speed off road navigation. The authors approaches the problem by combining semantic and geometric perspectives. This is done by lifting 2D RGB image features into 3D space using depth data. The stereo depth points are transformed into the RGB image to align resulting in a RGB-D image. The image is then improved using a self-supervised depth completion module that corrects and fills in gaps of the data to reduce errors. This is done with LiDAR data.

The model uses multiple BEV grids to represent different parts of the environment. The world is divided into a ground layer and a ceiling layer. The ground layer captures a semantic probability distribution, along with their highest and lowest points. The ceiling layer is similar to the ground layer but instead holds the information of the lowest hanging object. TerrainNet achieves the high inference speeds necessary for embedded systems while also maintaining geometric certainty in unstructured terrain.

2.5 The Development of Datasets in Unstructured Environments

Perception systems for autonomous driving has historically been driven by the availability of large datasets. Datasets such as KITTI [2], nuScenes [3] and Cityscapes [1] made advancements in on-road autonomy [4]. These scenes have a clear structure with road markings, roads and traffic patterns.

In unstructured off-road environments, the distinction between drivable and obsta-

cles becomes largely ambiguous due to problems with complex backgrounds, uncertain boundaries and lack of distinct drivable roads [11]. To deal with this, specialized datasets like RELLIS-3D [5] and The Great Outdoors [6]. These datasets typically combine RGB imagery with LiDAR point clouds [20]. Datasets like TartanDrive 2.0 [8] among others have further advanced the scope of off-road data engineering by providing additional sensor data streams within unstructured environments, for use in a wide variety of tasks.

While these advancements have generally extended the sensor modalilites, off-road datasets still generally suffer from limitations. The primary bottleneck lies in the manual annotation process [10]. Pixel perfect semantic segmentation in environments where boundaries are unclear is subjective, expensive and takes time. Whether it is dirt transitioning into grass or grass transitioning into mud. These datasets are often limited in scale and/or suffer from class imbalances. Large regions are often dominated by drivable road classes, while more hazardous classes occur much more infrequently. Because of the secretive nature of the autonomous UGV field, datasets are often not made public or are hidden behind research licenses.

Due to these limitations, scaling perception models using strictly supervised learning on limited, human-annotated datasets is challenging financially and practically [13]. The lack of annotated semantic data is a big motivation for the weakly supervised learning focus of this project. By the use of vision foundation models to automatically generate pseudo-ground truth, the dependence on manual annotation is removed or at the very least reduced.

2.6 Learning cost maps for offroad autonomous driving

In off road autonomous driving, cost maps are commonly applied together in the context of ROS for path planning. In the context of off road autonomy these cost maps are often constructed by hand. An approach that is slow and hard to fine tune particularly as the number of parameters grows. In a 2022 paper the authors propose CAMEL [21], a deep learning framework that learns cost map values from demonstrations, essentially construction cost maps that result in path as close as possible to the one that an expert driver has driven. CAMEL uses convolutional network to to generate these 2D cost-maps from a multi modal grid containing semantic projections into LiDAR point cloud, as well as LiDAR derived geometric information such as height and slope.

They then translate this costmap for the purpose of training into a navigation stack which produces steering and throttle commands to the robot. Training then intends to minimize the predicted inputs from the cost maps with the ones performed by the expert driver driving the same path.

2.7 RoadRunner M&M

RoadRunner M&M [14] is an end to end machine learning model for autonomous terrain navigation, developed at NASA jet propulsion laboratory together with ETH Zürich. The model takes RGB pictures from four cameras mounted on the vehicle, as well as LiDAR voxelmap as in data. Outputting in real time a traversability map and an elevation map in 2 different scales. One short range map with ±50m range and a 0.2m resolution as well as a longer range map at ±100m with a 0.8m resolution. This decision to have the network output maps at 2 different scales was done to allow for safer high speed autonomous navigation.

RoadRunner employed self supervised training in their creation of the ground truth, they utilize aggregated LiDAR point clouds and hindsight fusion to form a denser point cloud for each timestep which in turn is used to form BEV. To further refine the data they also utilize digital elevation maps from United States Geological Survey to further fill out the BEV in areas where the robot has not gathered sufficient information.

2.8 Lift Splat Shoot

Lift Splat Shoot [22] is an end to end 2D to BEV transformation model introduced in 2020. The authors propose a three stage pipeline transforming from perspective view to BEV space. The first stage of this pipeline, each pixels feature vector is expanded onto the 3D space through a depth distribution prediction which is learned implicitly from the ultimate task loss. The feature vector is then placed along the ray in 3D space with the feature vectoring being weighted according to the distribution. These are then projected onto the BEV grid by known camera intrinsics and extrinsics, aggregating all feature vectors by pillar pooling. A CNN then transforms this BEV feature grid into a grid containing semantic labels.

2.9 SimpleBEV

Unlike LSS, SimpleBEV approaches the problem of BEV creation differently. Rather than predicting a depth distribution in perspective view, SimpleBEV creates a fixed 3D voxel grid and using known camera intrinsics it projects the voxel grid into perspective view [23]. From perspective view it assigns the image features via bilinear interpolation to voxel space. The resulting 3D voxel feature space is then collapsed into BEV space and passed into a BEV convolutional decoder.

As a consequence of the architecture, the depth estimation dissapears as an explicit part that is learned. More practically this means that the unlike LSS the feature vector is assigned to every voxel in ray that the pixel maps to. This then means that BEV convolution decoder is trained to find coherent regions of the voxel in the BEV space and learn the BEV grid from this context. The resulting performance

of this style of architecture showed competitive performance compared to the older LSS style with explicit depth modules.

2.10 FocusBEV

FocusBEV is a monocular BEV segmentation framework whose central idea is a self-calibrated cycle view transformation [24]. A recurring problem in view transformation is that large portions of the perspective image are sky, distant background, and other regions that carry no useful information for the top-down grid yet still contaminate the BEV features during the transformation. FocusBEV handles this by performing the transformation as a cycle, after an initial perspective view to BEV mapping, the BEV features are projected back to perspective view and used to improve the image features, suppressing regions which do not contain ground level information before a second PV to BEV pass. Unlike LSS, the transformation is attention-based and does not rely on an explicit learned depth distribution. The full framework additionally includes an ego-motion-based temporal fusion module with a memory bank and an occupancy-agnostic IoU loss. In this thesis we adopt the cycle transformation as the defining component and refer to our single-frame variant as FocusBEV-inspired.

2.11 Semantic occupancy for autonomous vehicles

The use of self-supervised learning in autonomous vehicle context have been explored in a few papers but is still a rather unexplored area, especially in the off-road autonomous field. QueryOcc presents a self-supervised framework that predicts if pixels are occupied or free in an on-road environment with 4D input [25]. Data consists of continuous video streams and image features that are projected into 3D space. Additionally, all 3D images also has a timestamp meaning that each pixel has 4 features in total.

The model is provided with a query of these 4 features and tries to predict the occupancy of said pixel. The model is then provided with 4D images of past and future timestamps. This context gives the model more insight if the prediction of free or occupied space was correct or not. Then a foundation model is applied for semantic segmentation. To evaluate these occupancy predictions, QueryOcc uses RayIoU instead of standard voxel-based IoU. This is done by casting rays from the camera sensor and evaluating the predicted semantic labels along these rays.

2.12 Research Gap

In summary, big progress has been made in the off-road autonomy field, both in terms of 2D-to-3D projection architectures and off-road perception. However, one

large problem for further development is the systems reliance on human annotated training data or entirely unsupervised pipelines. The projection architectures reviewed here, LSS, SimpleBEV, FocusBEV, and BEVFormer, were all built and tested on large manually annotated on-road datasets. On the off-road side, OFFSEG and TerrainNet still lean on manual semantic labels, and while CAMEL and RoadRunner M&M manage to avoid them, they do so by swapping in expert demonstrations or purely geometric self-supervision. Literature evaluating how pseudo-labeled training data generated with zero-shot Vision-Language Models can replace or complement human annotated training data in the training of multi-class BEV models is limited. This thesis aims to fill this gap by reviewing the pseudo-labeled training data alternative and its impact on trained BEV models in unstructured off-road terrain.

3

Theory

3.1 The Pinhole Camera Model

The relationship between a 3D point in the camera frame, $P_c = [X_c, Y_c, Z_c]^T$, and the 2D pixel (u, v) in the image is given by the intrinsic matrix K. This matrix projects 3D coordinates onto the 2D image:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}.$$

Here, f_x and f_y are the focal lengths, and c_x and c_y are the image center points [26].

To combine images from different cameras, the points must be moved from the local camera frame to the vehicle frame. This is done with the extrinsic transformation matrix $T_{cam \to ego} \in \mathbb{R}^{4\times4}$:

$$\tilde{P}_{ego} = T_{cam \to ego} \tilde{P}_{cam}.$$

3.2 Bird's-Eye View Projection Methods

There are two main methods to create a BEV map from a normal camera image.

3.2.1 Forward Projection

Forward projection models first predict the depth of the image. For every pixel (u, v), the model predicts a depth distribution $\alpha(u, v) \in \mathbb{R}^D$ over D different depth values. It also predicts a feature vector $c(u, v) \in \mathbb{R}^C$. The feature is then moved into 3D space by multiplying these together:

$$f(u, v, d) = c(u, v) \cdot \alpha(u, v, d).$$

These 3D features are then moved into the vehicle's coordinate frame. They are placed into a 3D grid and then pushed down along the Z-axis to make a flat BEV map [22].

3.2.2 Backward Projection

Backward projection works in the opposite way. It starts by making an empty 3D grid in the vehicle frame [18] [27]. For each 3D point $V = \{x, y, z\}$ in this grid around the vehicle, the model calculates where that point would be in the 2D camera image. This uses the projection matrix $P = K[R|t]$. The matching 2D pixel (u, v) is found using this equation:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

The algorithm then looks at the 2D image features at pixel (u, v). It uses bilinear interpolation to read the feature and puts it back into the 3D grid. This method does not need to predict the depth of the image [28].

3.3 Deep Learning for Computer Vision

3.3.1 Transformer

The Transformer architecture was originally created for Natural Language Processing. The architecture introduced the self-attention mechanism [29]. Self-attention weighs the importance of different parts of the input data relative to one another. For an input sequence projected into Query (Q), Key (K), and Value (V) matrices, with the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V.$$

where d_k is the dimension of the key vectors. The Transformer architecture quickly became the standard architecture for complex sequence-to-sequence tasks.

3.3.2 Vision Transformer

Vision Transformers were introduced to apply the Transformer architecture to image recognition. Instead of processing pixels through convolutional filters, a ViT divides an image into separate patches [30]. Linear projection is used to create vector embeddings of the patches. The patches are then passed through the Transformer encoder block with a classification token and positional embeddings for each patch.

The ViT architecture is able to capture spatial relationships and representations across the full image due to its global receptive field. This makes Vision Transformers effective backbones for modern Vision Language Models, such as CLIP [31].

3.3.3 Mask2Former

Mask2Former changed how automatic segmentation tasks were handled [32]. Before this, segmentation was tackled per pixel. A class label was predicted for each

separate pixel. Mask2Former instead segments masks of the image. This means that pixels may be grouped and one label can be given to all the pixels of a masked region.

The difference of this architecture is masked attention within the decoder. In standard models, the network looks at every part of the image all at once. This takes a lot of computing power and can bring noise from the background. Masked attention fixes this by making the network only look at the mask area it found in the previous step. By ignoring the less important background noise, the model learns faster and draws much cleaner borders around objects.

This is done by adding an attention mask $\mathcal{M}$ at feature location (x, y) to the normal cross-attention equation:

$$\mathit{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \mathcal{M}\right)V.$$

The attention mask $\mathcal{M}$ is defined as:

$$\mathcal{M}(x, y) = \begin{cases} 0 & \text{if } (x, y) \in \text{mask} \\ -\infty & \text{otherwise} \end{cases}.$$

3.4 Vision-Language Foundation Models

Foundation Models are trained on massive scales of diverse data using self-supervised or weakly-supervised learning [33]. Vision-Language Model is a type of Foundation Model. By aligning visual features with natural language during pre-training, VLMs learn general-purpose representations.

An advantage with VLMs is their zero-shot or few-shot generalization. They encode structural and semantic understanding of images linked to human language. VLMs can be used in a various settings without the need for fine-tuning. By changing a text prompt, these models can dynamically adapt to identify new classes, shown to be effective for tasks such as pseudo labeling [34]. This zero-shot capability is the core of the pipeline in this project.

3.4.1 Zero-shot Semantic Segmentation

To gain understanding of input images from cameras, images can be divided into areas of semantic meaning. This is important and often used for models to be able to gain an understanding and recognize what different parts that an image consists of. This is crucial in many computer vision tasks and often used in autonomous vehicle systems. These datasets require each pixel of the images in the dataset to be categorized into classes described by a number. This type of data and labeling of images is called semantic segmentation. Each class is distinguished by their corresponding indices. Such datasets can be expensive and often require manual annotation where a human is set with the task of deducing the class of said pixel.

To mitigate the problem of manual annotation, modern approaches introduces the use of VLMs in order to semantically label images in a weakly-supervised manner. By the use of such models, the opportunity to create semantic segmentation pseudo ground-truth without any manual annotation arises. One way of doing this with custom classes is via zero-shot semantic segmentation. This means that the model is given instructions of what the class labels are together with a fitting description of what the model can expect to see in the image for each class. Therefore, to create an annotated semantic segmentation dataset, one needs to understand the distribution of semantic classes and include the different class categories and the explanation of each category in the instructions.

3.4.2 Contrastive Language-Image Pre-training

CLIP is a model used for zero-shot image recognition and was introduced in 2021 [31]. It builds upon a dual-encoder architecture which is trained on hundreds of millions of image-text pairs gathered from the internet. The model maps images and natural language text into latent space using a contrastive loss function. The cosine similarity between an image embedding I and a text embedding T is computed as:

$$sim(I, T) = \frac{I \cdot T}{||I||||T||}.$$

The visual representations learned by CLIP's image encoder have become the backbone for downstream prediction tasks for models performing zero-shot semantic segmentation.

3.4.3 Text and Image Prompted Segmentation

Models that build on CLIP used for zero-shot and one-shot image segmentation was later introduced. One such model is CLIPSeg which uses a lightweight transformer-based decoder that is given an arbitrary prompt with a describing text or an image [35]. The text prompt is encoded into an embedding vector and passed to the decoder via Feature-wise Linear Modulation. Given the feature map x and vector embedding of the text c, the FiLM layer transformation of features is defined as:

$$\text{FiLM}(x) = \gamma(c) \cdot x + \beta(c). \tag{3.1}$$

where $\gamma(c)$ and $\beta(c)$ are scaling and shifting parameters predicted from the text embedding using a multi-layer perceptron. In this way the network can adapt its attention and use the visual feature maps to segment parts matching the input text. With activations from CLIP encoder blocks and combined with the prompt via FiLM layers, the model produces the segmentation masks.

3.4.4 Cost Aggregation for Open-Vocabulary Segmentation

To further improve on segmentation architectures, models focusing more on cost aggregation were introduced. One such model is CAT-Seg which relies on image and text embeddings correlation rather than text to a decoder [36]. Instead of directly

optimizing raw feature embeddings, which often leads to severe overfitting on seen classes, the model computes the cosine similarity between the feature embeddings of image patches and text prompt. This rough semantic cost map is then refined through spatial and class-level aggregation modules. With this cost-aggregation approach, the network finetunes the CLIP model for dense prediction tasks. CAT-Seg have shown promising results in generalization on unseen classes.

3.5 Knowledge Distillation

Knowledge Distillation is a machine learning method designed to transfer knowledge from a complex, large "teacher" model to a more lightweight "student" model [37]. Given a teacher network T and a student network S, the feature distillation tries to minimize the distance between their respective feature maps, F^T and F^S. This is typically achieved using a distance metric such as Mean Squared Error or cosine similarity. In this thesis, KD is used to teach a computationally lightweight student model the complex semantic knowledge encoded by heavy foundation models.

3.6 DepthAnythingV2

DepthAnythingV2 is a model used to estimate monocular depth [38]. It uses a Dense Prediction Transformer with a DINOv2 encoder. The DPT takes features from the vision transformer and combines them using a convolutional decoder. This gives a depth value for every pixel in the image. This way, the model captures both local and global context at the same time.

DA2 is trained on a large dataset compromised of pseudo-labels. First, a large teacher model is trained on synthetic data. Then, this large model predicts the depth for 62 million real images without labels. After that, smaller student models are trained on both the synthetic data and the new pseudo-labels. This is referred to as knowledge distillation. DA2 can predict relative depth or metric depth.

In this thesis, we use the smaller ViT-Small model to find relative depth. We chose this because it runs fast enough on our hardware but still gives a good and dense depth map for every image.

3.7 RANSAC

Random Sample Consensus is an iterative algorithm for robust model fitting in the presence of outliers [39]. Methods like least-squares use all the data points which means that outliers can ruin the result. RANSAC instead draws a minimal random subset of data points, fits a model to that subset, and then counts how many of the remaining points fall within a threshold. Over many iterations, it keeps the model that has the most fitting points.

In this project, RANSAC is used to match the relative depth from DepthAnythingV2 to the real sparse distance from the LiDAR. We fit an affine model: $1/Z_{\text{metric}} = a \cdot Z_{\text{relative}} + b$. Here, a and b are parameters to estimate. Sometimes the LiDAR points are wrong because of bad calibration, dynamic objects, or reflections. RANSAC makes sure the outliers points do not ruin and corrupt the scaling of the final result. The result is a dense metric depth map suitable for accurate 3D unprojection.

This parametrization is used because the relative depth from DepthAnythingV2 is produced in a disparity-like (inverse-depth) space, so a global scale-and-shift relationship holds approximately between inverse metric depth and the relative prediction; fitting the affine model directly in metric-depth space would not correspond to the model's actual output and would systematically distort the alignment, particularly at long range.

3.8 Evaluation Metrics

The performance of semantic segmentation models is measured using IoU also known as Jaccard index. For a semantic class c, the IoU is defined as the ratio of the intersection of the predicted segmentation mask and the ground truth mask to their union with formula as:

$$\text{IoU}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}.$$

where TP_c as true positive, FP_c as false positive, and FN_c false negative predictions for class c.

To evaluate the performance of a model on all classes, the mIoU is calculated by taking the average of the class IoU defined as:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c.$$

where C is the total number of classes. While mIoU gives a summary of model performance, class IoU is often be more important in off-road navigation. Not detecting a rare dangerous class can have far worse consequences than not detecting all parts of the background terrain. The matter in how it missclassifies terrain is also interesting, for example if obstacles labaled as free space could be devestating while driving, while an obstacle classified as a human is likely inconsequential for traversing terrain.

To contextualize the reported mIoU, two label-free reference baselines are computed on each test set. The constant baseline assigns every valid BEV cell to the free-space class, the naive drivable everywhere policy.. The prior-frequency baseline assigns each cell a class drawn at random in proportion to the test-set class frequencies; under independent assignment its expected per-class IoU reduces to $IoU_c = p_c/(2 - p_c)$, where p_c is the frequency of class c.

4

Methods

4.1 System Overview

The pipeline uses multimodal sensor streams in order to create a BEV representation of the local surroundings of the vehicle. Manually labeled datasets of off-road environments are expensive, and limited in scalability which is why this pipeline consists of a weakly-supervised framework. The architecture builds upon a sort of knowledge distillation which is why we divide the architecture into branches "student" and "teacher".

The teacher branch which uses vision language models and helps construct the pseudo-ground truth. This branch consists of two teachers.

- **Semantic Teacher:** Uses CAT-Seg to generate dense, per-pixel semantic labels across a customized off-road classes for all frames.

- **Depth Teacher:** Relies directly on LiDAR sweeps in combination with depth estimation network. The network is used to fill the gaps of the LiDAR sweeps to increase the density of depth points used when projecting in BEV space.

The teacher branch is how features of the images are distilled into our training targets. The student branch consists of architectures with lower complexity aiming to learn the representation of the teacher models for a lower computational cost.

The goal of the student network is to distill the knowledge from the teacher. During the weakly-supervised training phase, the student processes its inputs and projects its own learned features into a local BEV grid. Its predictions are then improved by comparing them against the targets generated by the offline teacher pipeline. This allows the autonomous system to use the student model during deployment. Figure 4.1 shows the full pipeline architecture, showing the teacher and student branches and how targets are created and used during training.

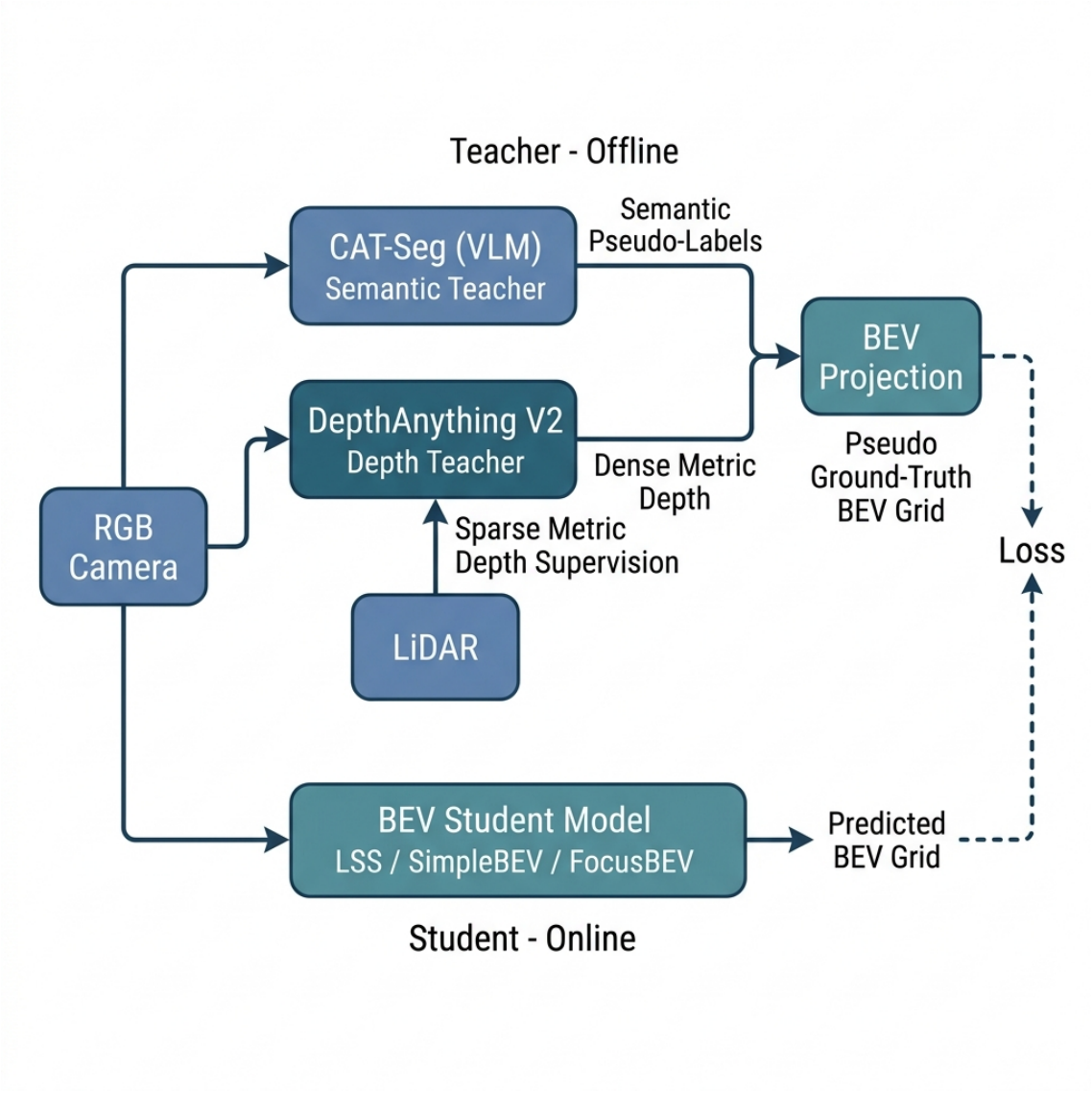


Figure 4.1: Overview of the teacher-student pipeline. The offline teacher branch uses CAT-Seg and DepthAnything V2 together with LiDAR supervision to create pseudo ground-truth BEV grids. The online student branch learns to predict the BEV grid only using the RGB as input.

4.2 Data Acquisition and Preprocessing

To capture this environment, our system uses data from sensors such as three RGB cameras, LiDAR, odometry and poses. Before this data can be fused, it must be aligned in the time domain. This section details the datasets used and the synchronization strategies used to map the LiDAR sweeps to the correct camera frame timestamps. For training and testing we use the dataset The Great Outdoors [6]. For out-of-distribution testing we use the dataset Rellis-3D [5], a dataset constructed by the same organization as The Great Outdoors dataset.

Two datasets are used in this work: The Great Outdoors [6] for in-distribution training and testing, and RELLIS-3D [5] for out-of-distribution evaluation. Both datasets were published by the same organization, which ensures a large overlap in class definitions and annotation conventions. Table 4.1 summarizes the key characteristics of each dataset.

The Great Outdoors originally contains approximately 55,000 frames recorded at high frequency. Because the LIDAR was only sampled at one third the rate, approximately two thirds of the frames were filtered out by sub-sampling the nearest camera frame to each LiDAR scan (with the largest time deviation being 8ms between RGB image and LiDAR scan), yielding roughly 18,000 unique frames used for pseudo-label generation and BEV target construction. Of these, approximately 1,000 frames carry manual annotations and are split into train, validation, and test sets at a 70/15/15 ratio. The approximately 17,000 remaining unannotated frames form the pseudo-labeled training set. In addition, to avoid spatial leakage between the different sets, each set had the last five seconds of data removed.

	The Great Outdoors	**RELLIS-3D**
Role	In-distribution	Out-of-distribution
Raw frames	≈55,000	≈6,000
After filtering	≈18,000	≈6000
Annotated frames	≈1,000	≈6,000
Train / Val / Test	70 / 15 / 15 %	Test only
Original classes	22	20
Collapsed classes	5	5
Sensors	3× RGB, LiDAR, IMU	1× RGB, LiDAR, IMU

Table 4.1: Dataset statistics for the two evaluation datasets. The great outdoors dataset includes frames from three cameras of which one is used in this project.

4.3 Semantic Representation using CAT-Seg

To establish the semantic layout of the environment, the pipeline uses CAT-Seg, a zero-shot image segmentation model capable of generating dense prediction masks from arbitrary text prompts [36]. Rather than training a specialized annotator, we query the pre-trained VLM with 5 custom class categories for UGV path planning. This is due to two reasons. Firstly, in the context of off-road vehicles, obstacles such as rocks and trees pose similar threat to the system. Sizes can vary but this does not have to do with semantics. Similarly different types of roads such as gravel, asphalt, etc, can be treated more broadly as drivable surfaces. Secondly, for a fair comparison between pretrained models and models trained on pseudo data, the class labels need to be unified which is possible when labels are broad by mapping the more narrow classes. The classes used by the pipeline are the following: Free space which includes different roads, high cost including more costly surfaces, obstacle which groups obstacles, dynamic for moving objects and void for empty space.

Class Label	Sub labels
`free_space.`	dirt road, gravel, grass, concrete, asphalt road, mulch
`high_cost`	mud, puddle, body of water, rubble
`obstacle`	tree, bush, building, wall, fence, utility pole, barrier, tree log
`dynamic`	person, vehicle
`void_ignore`	sky, background void

Table 4.2: CAT-Seg Semantic Class Prompts.

By processing the camera frames through CAT-Seg, the pipeline estimates the true semantic labels of the scene by producing the semantic pseudo-labels. These labels are used to train models to understand the semantic parts of image input received through cameras. However, fully understanding local surroundings also requires an understanding of the size and distance to things. This information cannot be found reliably in simple 2D monocular images. That is why depth data is also necessary for a model since this requires the combination of geometric and semantic understanding.

4.4 LiDAR Supervision

Before building a target BEV grids we perform a 3 step LiDAR supervised depth pipeline in 2D image space. First, we use monocular depth estimation model DepthAnythingV2 built on ViT-small backbone to generate a unitless depth map of each RGB image. Then the LiDAR pointcloud is projected into perspective view using known extrinsics and intrinsics, giving sparse metric depth to a subset of pixels in image space. Finally, the sparse metric LiDAR depth and the dense relative depth from DepthAnything V2 are fused using a RANSAC-based alignment. An affine model $\frac{1}{Z_{\text{metric}}} = a \cdot Z_{\text{relative}} + b$ is fitted in disparity space between the sparse LiDAR measurements and the corresponding relative depth values. RANSAC rejects outliers caused by calibration errors or dynamic objects, producing robust scale and shift parameters. The dense metric depth map is then used for projection. Figure 4.2 summarises the three-step depth pipeline.

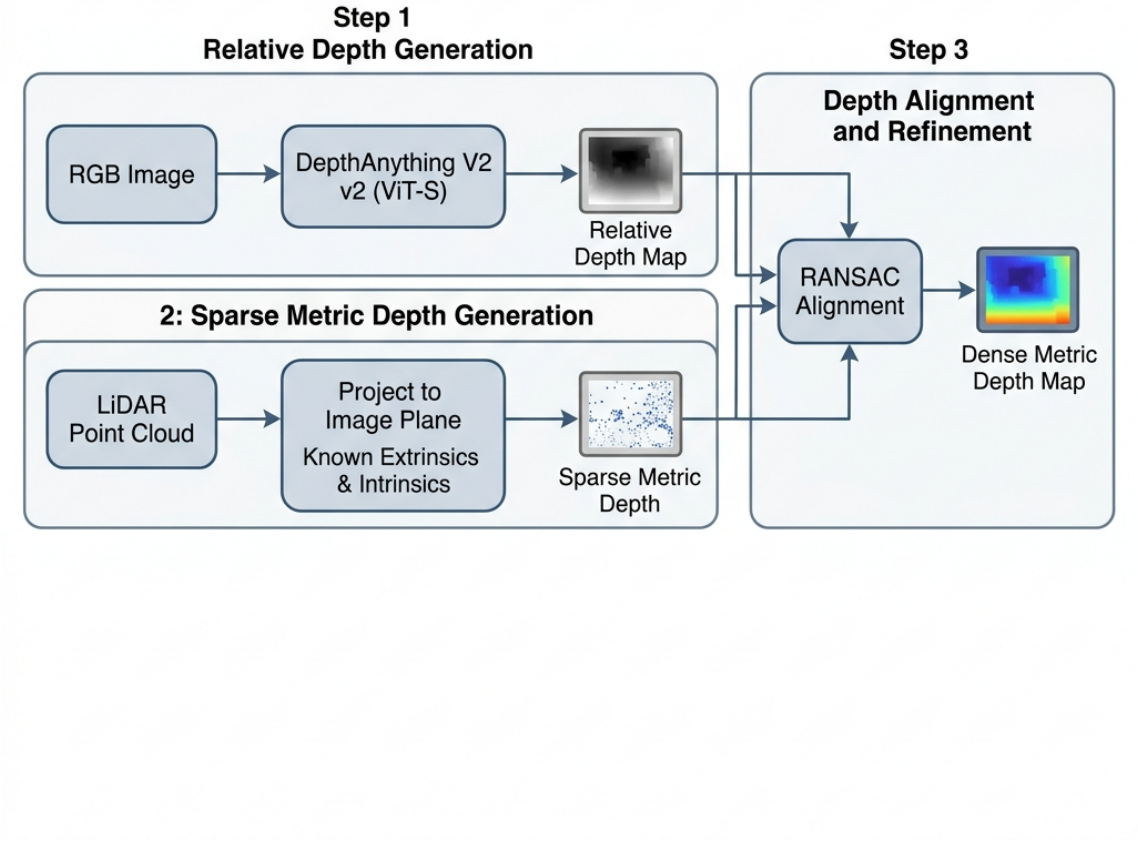


Figure 4.2: The three-step LiDAR-supervised depth pipeline. Step 1 generates a relative depth map from RGB using DepthAnything V2. Step 2 projects the LiDAR point cloud into image space to obtain sparse metric depth. Step 3 aligns the two using RANSAC to produce dense metric depth.

4.5 Bird's-Eye View Projection

The BEV representation serves as a useful spatial framework for a UGV's local path planning. By projecting the 3D environment into a 2D top-down grid common in autonomous driving tasks, the navigation system can more efficiently evaluate local traversability. The BEV grid is configured with a resolution of 100×100 cells, capturing a physical area of 25 meters forward and 25 meters laterally relative to the ego-vehicle. Regions within the grid but outside the fov of the camera are not included and are labeled as non valid cells in the grid. The decision to use a 100x100 grid was motivated in the thought it would better capture smaller obstacles in BEV

space. It would lessen the chance that other more common classes would drown out the obstacle class in these cases.

4.5.1 Coordinate Transformations and Unprojection

To populate the BEV grid, 2D camera pixels and dense depth maps must first be unprojected into 3D space. Given a pixel coordinate (u, v) and its corresponding metric depth Z_c, the 3D coordinates in the camera frame, $P_{\text{cam}} = [X_c, Y_c, Z_c]^T$, are obtained using the camera intrinsic matrix K:

$$P_{\text{cam}} = Z_c K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = Z_c \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{4.1}$$

where f_x, f_y are the focal lengths, and c_x, c_y represent the principal point.

The points are then transformed into the vehicle's ego frame using the homogeneous extrinsic transformation matrix $P_{\text{cam}\to\text{ego}} \in \mathbb{R}^{4\times4}$:

$$\tilde{P}_{\text{ego}} = P_{\text{cam}\to\text{ego}} \tilde{P}_{\text{cam}}. \tag{4.2}$$

4.5.2 Depth Integration

To populate the BEV grid, the dense metric depth maps produced by the three-step LiDAR-supervised pipeline are used. Each frame is processed independently, every pixel is unprojected to 3D using its dense metric depth value and the camera intrinsics, then discretised into the BEV grid. This avoids the need for temporal aggregation or explicit sensor fusion during BEV construction, while still benefiting from the LiDAR-aligned metric depth.

4.5.3 BEV Ground Truth Generation

Once the 3D points have been unprojected from the dense depth map, they are discretized into the BEV grid. For each valid cell (r, c), spatial and semantic features are extracted to form pseudo-ground truth targets. The pseudo-labels generated by the larger teacher models are mapped to the 3D points. In each BEV cell, the semantic classes from CAT-Seg are aggregated using a majority vote. CAT-Seg provides the necessary semantic vocabulary. Semantic segmentation accuracy is evaluated using mIoU and class-wise IoU metrics.

Finally, to resolve empty cells caused by sensor occlusions and sparse depth points at far ranges, nearest-neighbor infill is applied within a fixed radius. For each invalid cell within the radius, the value of the nearest valid cell is assigned. The infill radius was set at seven, this value was a result of a qualitative analysis of the resulting target BEV and their corresponding images. Where clear gaps and streaks were observed at longer range in BEV when the infill radius was set to a lower value, it also served a secondary purpose to fill in regions which were occluded but could reasonably be assumed to be of the same class type in BEV space.

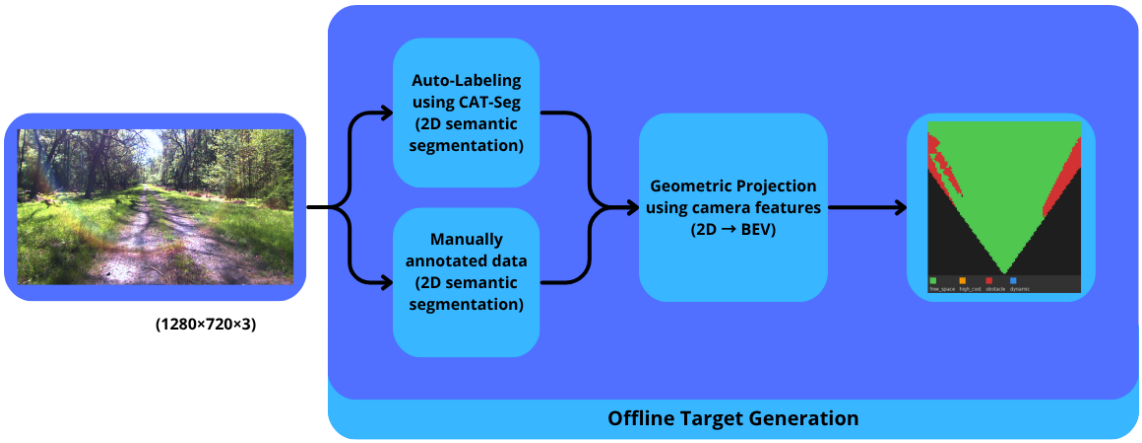


Figure 4.3: Simplified figure over semantic BEV target generation. Image from "The great outdoors dataset" by Peng Jiang et al., 2025. Licensed under CC BY-NC-SA.

4.6 Semi-Supervised and Pseudo-Labeling Framework

The methodology compares fully supervised, weakly-supervised, and semi-supervised training approach to determine the optimal strategy for semantic traversability mapping.

The goal is to train and benchmark BEV architectures across three distinct training tracks, varying both the volume and the quality of the supervisory data.

To evaluate the impact of dataset scale and annotation quality, the training pipeline is divided into three primary tracks.

- **Supervised Training with Annotated Data:** This track acts as the performance baseline by training the models only on the hand-annotated ground truth dataset. This track evaluates the performance of models trained using limited manually labeled data.
- **Weakly-Supervised Training with Pseudo-Labeled Data:** To use the full scale of the dataset without requiring human input, this track uses machine-generated pseudo-labels. The zero-shot vision-language model, CAT-Seg, is used to auto-label the image data with semantic classes for each pixel. Models are trained purely on these pseudo-labels. This track evaluates whether the volume of weakly-supervised data can compensate for the noise and boundary inaccuracies of the VLM's predictions.
- **Semi-supervised with Hybrid:** The final track combines the previous two approaches into a semi-supervised framework. It fine-tunes the models trained on psuedo-labeled data with the human-annotated dataset. The goal is to evaluate whether models trained on a larger pseudo-labeled dataset can be complemented with a smaller manually annotated dataset.

By evaluating the models trained using these methods, this framework can give useful insights into the trade-offs between manual annotation and automated dataset scaling in UGV perception.

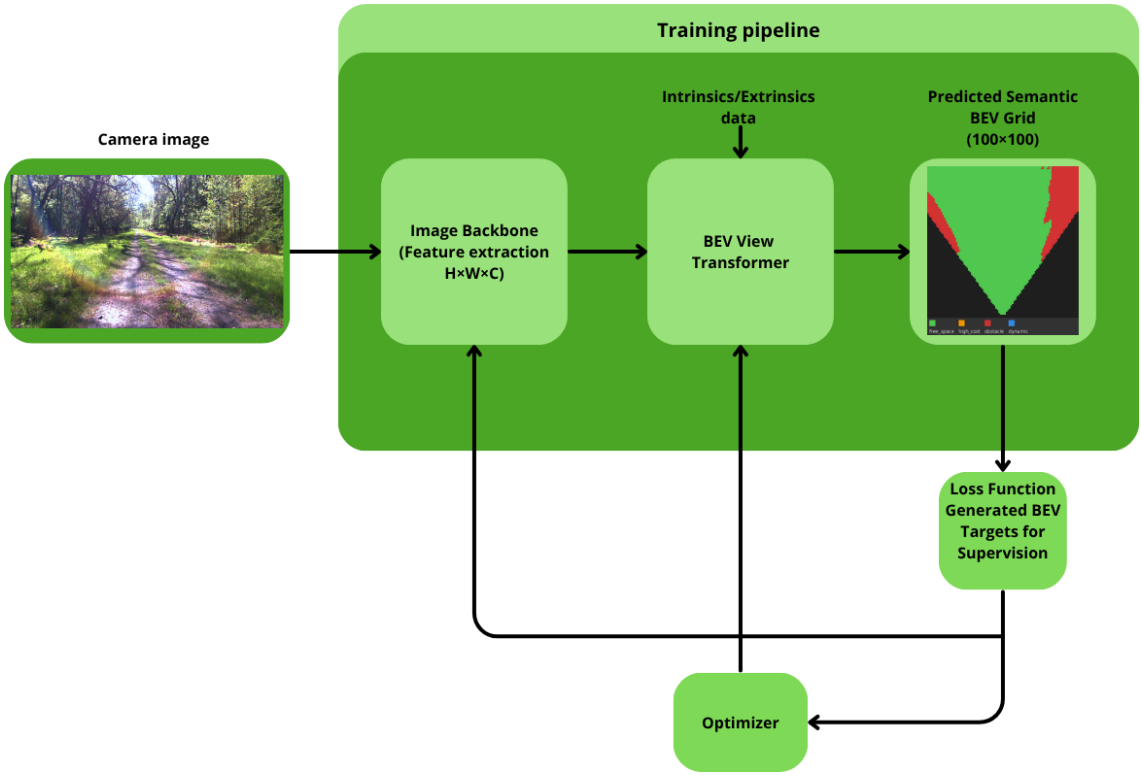


Figure 4.4: Simplified figure over training BEV models. Image from "The great outdoors dataset" by Peng Jiang et al., 2025. Licensed under CC BY-NC-SA.

4.7 Evaluated BEV architectures

Rather than designing a custom neural network architecture, the methodology of this thesis focuses on creating an autonomous pipeline for dataset preparation to train and evaluate several BEV architectures on the generated UGV dataset. The primary objective is to benchmark how different 2D-to-3D BEV frameworks perform when tasked with estimating semantic BEV maps for off-road navigation and comparing the different training approaches.

While all models compared share the same input/output formulation, it is important to note that this is a comparison between practical implementations rather than an isolated study of projection mechanisms. The architectures differ in how they project the 2D features into 3D, but they also retain specific backbone choices and loss formulations from their original implementations.

4.7.1 Lift, Splat, Shoot

The Lift, Splat, Shoot framework is our baseline for forward-projection architectures. LSS models depth uncertainty.

- **Lift:** The network predicts a class distribution over D predefined depth bins for each pixel in the 2D feature map. The image features are then computed as an outer product with these depth probabilities to lift them into 3D.
- **Splat:** The features from the camera are transformed into the UGV's local coordinate frame using extrinsic matrices. They are then assigned to a 3D voxel grid and "splatted" using sum pooling to form a 3D representation.
- **Shoot:** The Z-axis is collapsed to form the 2D BEV plane. This is then processed by a convolutional task head to create the terrain.

LSS encodes depth uncertainty directly into the feature representation making it robust. However, the forward-projection mechanism can lead to sparse artifacts or "shadows" in the resulting BEV grid, particularly at longer ranges where the frustum volume expands.

4.7.2 SimpleBEV

SimpleBEV takes the opposite approach as an inverse-mapping method. Instead of predicting depth to push features into 3D space, SimpleBEV pulls features directly from the 2D image plane into a predefined 3D grid [23]

- **Voxel Definition:** The architecture begins by defining a volume surrounding the UGV.
- **Inverse Sampling:** For every voxel in this 3D grid, its coordinate (X, Y, Z) is transformed into the camera frame and projected onto the 2D image plane using the intrinsic and extrinsic matrices. The network then samples the 2D feature map at that exact projection point using bilinear interpolation.
- **Feature Aggregation:** Each voxel receives at most one feature sample. The populated 3D grid is then collapsed into the BEV plane by joining the Z-slices on the channel dimension.

SimpleBEV entirely bypasses the need for a discrete depth prediction head, making it simpler in design and free from the sparsity problems seen in forward-projection models like LSS. However, because it assumes the 3D geometry is known it relies heavily on accurate extrinsic calibration and dense feature representations.

4.7.3 FocusBEV

FocusBEV's central contribution is a cycle view transformation, rather than lifting features once from perspective view to BEV, it projects a second time, feeding information from the initial BEV estimate back into the PV feature map before repeating the lift. This back-and-forth refinement is intended to improve spatial alignment between the two domains.

- **Feature Extraction:** An EfficientNet-B0 backbone extracts image features at stride 16, projected to 192 channels, with each spatial position augmented by a positional encoding derived from the per-pixel 3D ray direction computed from K.
- **Cycle View Transformation:** Learned BEV queries attend to the PV feature map via multi-head cross-attention to produce an initial BEV grid $P_{\text{bev}}^{(1)}$. These features are projected back to PV-space where ray-encoded queries produce a refinement signal P_{refine}, which is fused with the original PV features as $P'_{\text{pv}} = \text{Conv}([P_{\text{pv}},\ \alpha \cdot P_{\text{refine}}])$. The refined features are lifted to BEV a second time using shared weights, giving $P_{\text{bev}}^{(2)}$, and the two grids are fused into $P_{\text{bev}} = \text{Conv}([P_{\text{bev}}^{(1)},\ P_{\text{bev}}^{(2)}])$.
- **Prediction Head:** A lightweight convolutional head maps the final BEV features to per-cell semantic class logits.

Our implementation follows this cycle view transformation but omits the original framework's ego-motion temporal fusion and occupancy-agnostic IoU loss, and is therefore referred to as FocusBEV-inspired. The α gate is held at zero for the first three epochs and linearly ramped to one over the following five, letting the backbone stabilize before cycle refinement is introduced.

4.7.4 Implementation Constraints and Standardization

To get a good level of comparability, all three architectures share the same input and output formulation: a single RGB image is processed and a 100×100 BEV semantic grid is produced. However, to evaluate the models as they were originally designed, their underlying configurations are maintained. The backbones differ between architectures: LSS and FocusBEV both use EfficientNet-B0 as the image encoder, while SimpleBEV uses a ResNet-50 backbone following its original implementation. By keeping these differences, we are comparing the full models as they were originally built, rather than just testing their projection methods in isolation.

4.8 Training Objective and Loss Functions

The primary goal of training the models is to minimize the difference between the predicted BEV semantic map of the models and the target ground truth. Because the network performs dense pixel-wise classification over the BEV grid, the problem is formulated as a multi-class semantic segmentation task.

However, off-road autonomous navigation environments often suffer from severe class imbalance, where large parts of the datasets are covered by common safe classes that outnumber the classes of critical obstacles [5] [40] [6]. Large regions of the BEV grid are typically dominated by things that would belong to the free space category in this thesis. Other hazards that an autonomous system might face would be found in the obstacle class, or dynamic class and occupy only a small fraction of the area. To prevent the network from converging to majority-class predictions, certain metrics

can be used to weigh different classes according to the class distribution of the dataset [41]. Thus, a composite loss function is employed.

4.8.1 Weighted Cross-Entropy Loss

The main loss term is the Weighted Cross-Entropy loss. For a given BEV grid cell i, let y_i represent the one-hot encoded ground truth label and $\hat{y}_i$ represent the predicted softmax probability distribution across the C terrain classes. The WCE loss is defined as:

$$\mathcal{L}_{WCE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot y_{i,c}\log(\hat{y}_{i,c}). \tag{4.3}$$

where N is the total number of valid pixels in the BEV grid. The class weights w_c are calculated to match the class frequencies within the training dataset. This forces the network to penalize misclassifications of rare, high-risk classes more heavily than misclassifications of common classes.

4.8.2 Dice Loss

WCE only looks at individual pixels, so it doesn't care whether the predictions form clean, connected regions. To handle class imbalance and encourage cleaner boundaries between classes, a Dice Loss is also used. The Dice Loss maximizes the IoU between the predicted and target masks:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i\hat{y}_i + \epsilon}{\sum_{i=1}^{N} y_i + \sum_{i=1}^{N}\hat{y}_i + \epsilon}. \tag{4.4}$$

where ϵ is a small smoothing constant to prevent division by zero. By optimizing the overlap, the Dice Loss keeps small objects from fading into surrounding background classes.

4.8.3 Void Masking

One issue with BEV projection is handling areas that fall outside the camera's view. Because camera frustum cannot observe the entire 360-degree environment evenly, certain areas of the 3D grid are occluded, out of sensor range, or project to the sky.

These regions are mapped to a *void_ignore* class in the ground truth. During the forward pass, a binary mask is dynamically generated to exclude any grid cell labeled as *void_ignore* from the loss calculation. This masking helps make gradients are backpropagate from valid, observable terrain, preventing the network from wasting capacity guessing the contents of void space.

The loss formulation is applied per architecture. FocusBEV uses both terms: $\mathcal{L}_{total} = \lambda_1\mathcal{L}_{WCE} + \lambda_2\mathcal{L}_{Dice}$, where λ_1, λ_2 are set weights. LSS uses weighted cross-entropy loss but no Dice loss. SimpleBEV uses weighted cross-entropy only. While the differences in objective functions together with differing backbones makes for a less strict isolated comparison between models, this is intended for the architectures to work at their best capabilities. The goal of the thesis is not a pure architecture

comparison, rather a exploration of the implications of pseudo-labeled data used to train the three different architectures.

4.9 Implementation Details

This section outlines the specific hardware, software frameworks, and hyperparameters utilized to train the BEV models and generate the pseudo-ground truth targets.

4.9.1 Software Frameworks

The pipeline was implemented in Python. The deep learning parts of the project were developed using the PyTorch framework. For the semantic pseudo-labeling, CAT-Seg [36] was used as the zero-shot segmentation model, built on top of Detectron2 and the Mask2Former architecture [42] [32]. The base model using the ViT-B/16 variant with an input resolution of 384×384 was chosen over the larger ViT-L and ViT-H variants due to the computational limits of the available hardware. The larger models exceeded the VRAM available on the L4 GPU when processing full-resolution frames across the dataset. Hugging Face Transformers and OpenCLIP were used to load the ViT backbones and process the text embeddings [43]. For monocular depth estimation, DepthAnythingV2 [38] was used with the ViT-Small backbone.

4.9.2 Hardware Specifications

Given the computational requirements of processing images, dense point clouds, and large foundation models, all experiments were made on a high-performance SLURM computing cluster. Both the pseudo-label generation script and the BEV training loops were executed on compute nodes with one L4 GPU.

4.9.3 Pseudo-Ground Truth Generation

To generate $\approx$ 18.000 semantic targets, CAT-Seg was used to process the images. The model received the text dictionary mapping of the five broad classes. Each sub-label was formatted using the base prompt template "You see a {label} in the image" before being passed to the text encoder. The logits from sub-prompts were aggregated using a maximum-confidence pooling strategy to form the final discrete semantic maps, which were saved as targets for the images.

4.9.4 Model Training and Hyperparameters

The BEV models were trained end-to-end to predict the terrain cost maps, with the goal of keeping the core essence of each model as implemented in their actual paper. Given our computational capabilities, they were trained with the following configurations:

Parameter	FocusBEV	LSS	SimpleBEV
Epochs	20	20	20
Batch size	4	4	4
Optimizer	AdamW	Adam	AdamW
Base LR	2×10^{-4}	1×10^{-3}	3×10^{-4}
Weight decay	1×10^{-4}	1×10^{-7}	1×10^{-5}
LR Scheduler	Cosine Annealing	Step ($\times$0.1 / 20 ep)	Cosine Annealing

Table 4.3: Training configurations per architecture.

From testing we arrived at the stated number of epochs as seen in table 4.3 to be 20 as no improvement of either of the three models were observed afterwards.

5

Results

5.1 Performance of Pseudo-labeling

The segmentation model CAT-Seg was used to generate segmentation masks on all images from the annotated dataset not included in the test set. The train, val, test split of 70/15/15 on the annotated dataset. The following statistics comes from the comparison of ≈ 1k annotated and segmentation masks on 2D RGB images.

Free space	High cost	Obstacle	Dynamic	Void ignore	**mIoU**
0.904	0.063	0.873	0.282	0.579	0.540

Table 5.1: Class-wise IoU and mIoU between manually annotated and pseudo masks.

Class	**Precision**	**Recall**	**F1-score**	**Support**
Free space	0.952	0.948	0.950	562.522
High cost	0.069	0.430	0.119	2.025
Obstacle	0.941	0.923	0.932	365.127
Dynamic	0.688	0.323	0.439	0.827
Void ignore	0.763	0.706	0.733	8.610

Table 5.2: Classification Metrics by Class. Support scaled by 10^{-6}.

Figure 5.1 visualizes the class IoU between the pseudo-labels generated by CAT-Seg and the manual annotations. Free space and Obstacle are captured well. High cost is almost entirely missed and Dynamic suffers from low recall. The low accuracy on these classes can affect the student models which are trained on this ground truth.

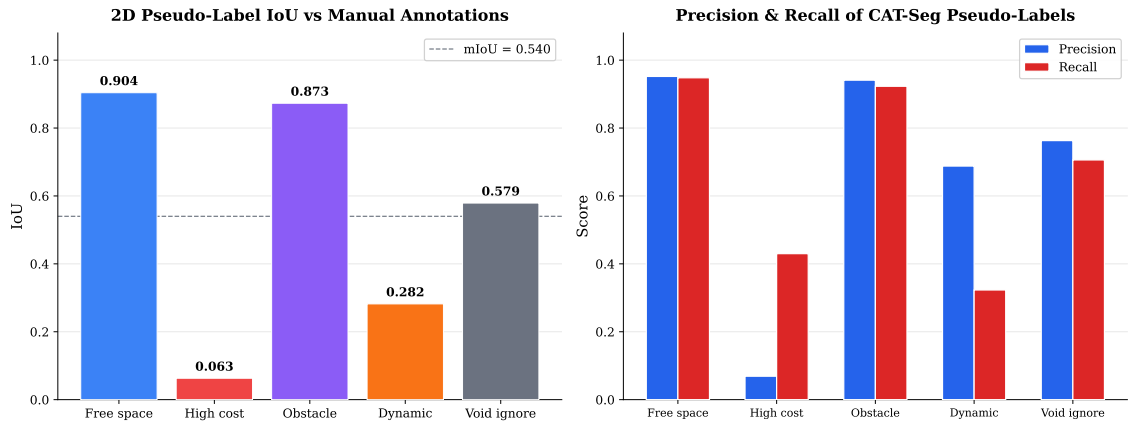


Figure 5.1: Quality of the 2D pseudo-labels produced by CAT-Seg. Class IoU against manual annotations. Per-class precision and recall.

There is an extreme class imbalance in the training set. As shown in Figure 5.2, Free space and Obstacle together account for over 98% of all labeled pixels, while High cost and Dynamic only account for about 0.3% together. The imbalance ratio between the most occurring class and least is roughly 680:1. This makes it very difficult for both the pseudo-labeling model and the BEV student networks to learn the minority classes.

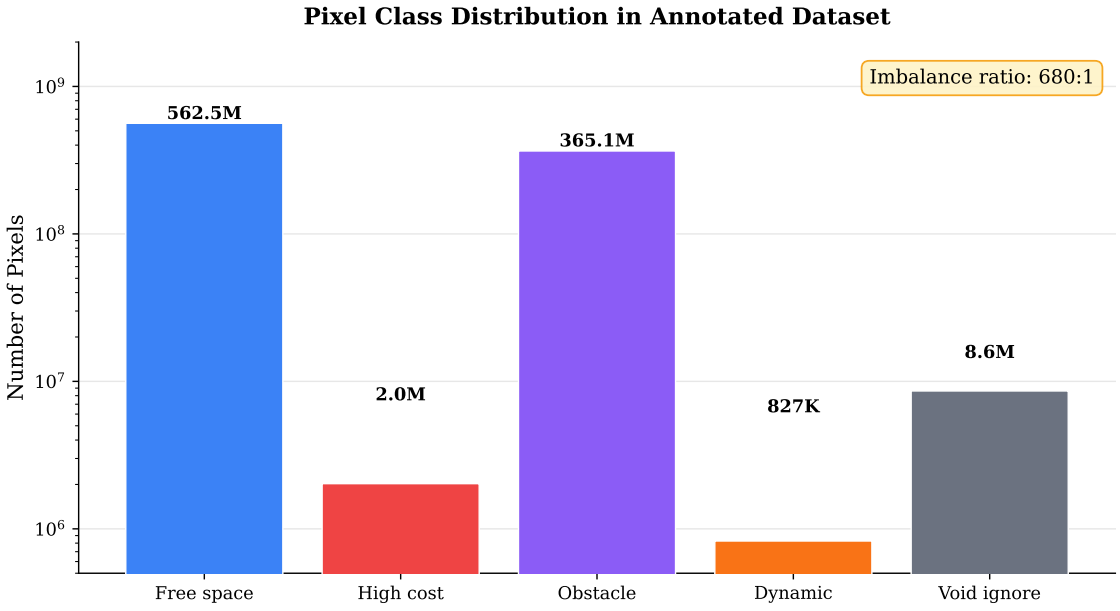


Figure 5.2: Pixel-level class distribution in the annotated dataset. The 680:1 imbalance between the majority and minority classes explains the near-zero IoU for underrepresented categories.

This class imbalance is further visible in the generated BEV targets from the annotated dataset as noted in table 5.3.

Class	Share of grids (%)
Free Space	87.44
High Cost	0.20
Obstacle	12.32
Dynamic	0.05
Valid points	7,715,884

Table 5.3: Per-class grid share (%) of annotated BEV ground-truth. Only valid points in the grid are counted, void grid points are excluded.

5.2 Model Evaluation

Because all results in this chapter come from single training runs, we have no per-configuration variance estimates and do not perform significance testing (see §6.6). To avoid over-reading small differences, we adopt a conservative interpretability band: differences in mIoU below roughly 0.02 are treated as within the range plausibly attributable to run-to-run variation and are not interpreted directionally, while differences of 0.02 or more are interpreted, with the largest gaps ($\geq$0.04) treated as the most secure. This is a heuristic magnitude threshold, not a statistical bound. It is set deliberately wide relative to the low run-to-run variation in validation loss and test-set mIoU informally observed across development runs, so that only differences clearly exceeding that scale are given weight.

5.2.1 Training on Manually Annotated Ground-Truth

Models were trained and validated on the BEV targets constructed using the train and val sets of the manually annotated data. Models were then tested on BEV targets constructed using the test portion of the manually annotated semantic data.

Model	Free space	High cost	Obstacle	Dynamic	mIoU
LSS	0.918	0	0.309	0	0.307
SimpleBEV	0.935	0	**0.505**	0	**0.360**
FocusBEV	**0.936**	0	0.429	0	0.341

Table 5.4: Class-wise IoU and mIoU on manually annotated BEV targets.

5.2.2 Training on Pseudo Ground-Truth

Models were trained and validated on the BEV targets constructed using the train and val sets of the pseudo-labeled data. Models were then tested on BEV targets constructed using the test portion of the manually annotated semantic data.

Model	Free space	High cost	Obstacle	Dynamic	**mIoU**
LSS	0.824	0	0.191	0	0.254
SimpleBEV	0.913	0	0.391	0	0.326
FocusBEV	**0.917**	0	**0.478**	0	**0.349**

Table 5.5: Class-wise IoU and mIoU on pseudo-labeled BEV targets.

Relative to the annotated baseline (Table **??**), the LSS drop (0.307 to 0.254) and the SimpleBEV drop (0.360 to 0.326) exceed the 0.02 interpretability band, while the FocusBEV change (0.341 to 0.349) falls within it and is treated as a tie.

5.2.3 Fine-tuning Models

The models trained using the pseudo-labeled dataset were then fine-tuned using the train/val data split from the annotated dataset. Here the hybrid approach is evaluated, were BEV GT constructed on annotated data and on pseudo-labeled data is used during the training phase.

Model	Free space	High cost	Obstacle	Dynamic	**mIoU**
LSS	0.854	0	0.222	**0.001**	0.269
SimpleBEV	0.935	0	0.508	0	0.361
FocusBEV	**0.943**	0	**0.531**	0	**0.369**

Table 5.6: Class-wise IoU and mIoU for hybrid training.

Among the hybrid results, the LSS drop from the annotated baseline (0.307 to 0.269) and the FocusBEV gain (0.341 to 0.369) exceed the band, the SimpleBEV change (0.360 to 0.361) falls within it.

5.2.4 Out-of-Distribution Test

All the model and training method combinations trained on the BEV prediction task were then further evaluated using a OOD test set to get a better picture on the models performance and ability to generalize.

	Model	Free space	High cost	Obstacle	Dynamic	mIoU
Pseudo	FocusBEV	0.434	0	0.359	0.004	0.199
	LSS	0.370	**0.025**	0.446	**0.016**	0.214
	SimpleBEV	**0.438**	0.001	**0.560**	0.003	**0.250**
Manual	FocusBEV	**0.419**	0	0.184	0	0.151
	LSS	0.400	0	0.105	**0.059**	0.141
	SimpleBEV	0.385	0	**0.565**	0	**0.238**
Hybrid	FocusBEV	**0.483**	0.000	0.491	0.000	0.244
	LSS	0.357	**0.027**	0.224	0.018	0.157
	SimpleBEV	0.430	0.000	**0.582**	**0.036**	**0.262**

Table 5.7: Class-wise IoU and mIoU trained on pseudo, annotated, and hybrid labels, tested on OOD.

Figure 5.3 presents the confusion matrices for SimpleBEV across all three training methods on the in-distribution test set. Annotated training shows the highest performance with free-space recall at 96.4% and the highest obstacle recall at 69.0%. Pseudo-label training struggles by comparison, dropping obstacle recall down to 58.4%. Hybrid training performs comparably to the Annotated baseline in-distribution.

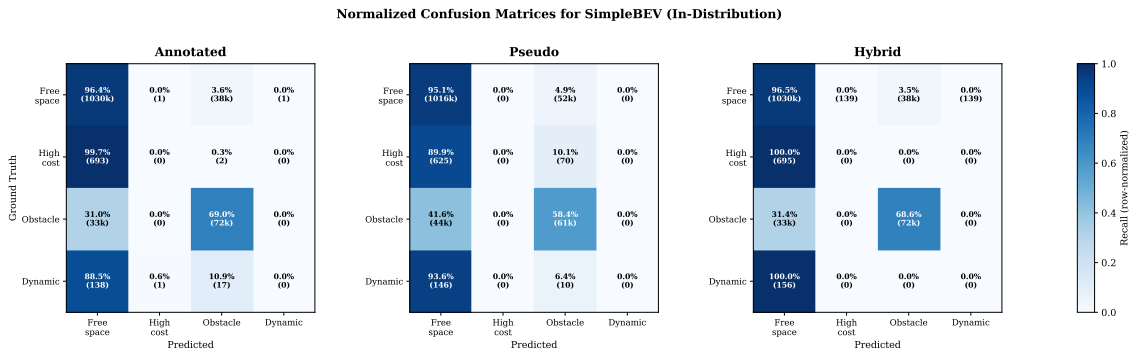


Figure 5.3: Confusion matrices for SimpleBEV on the in-distribution test set. Each cell shows the recall percentage and absolute pixel count.

Figure 5.4 shows the corresponding confusion matrices on the OOD test set, to show differences. While annotated training showed the strongest performance on in-distribution data, it shows over-sensitivity in out-of-distribution environments, misclassifying nearly 43% of free space as obstacles. Pseudo labels performs better in the new environment and reduces these obstacle hallucinations by improving recall from 57.2% to 69.4%. Annotated has better obstacle recall when compared with pseudo but is also overly confident in those predictions. Pseudo shows higher precision and stability.

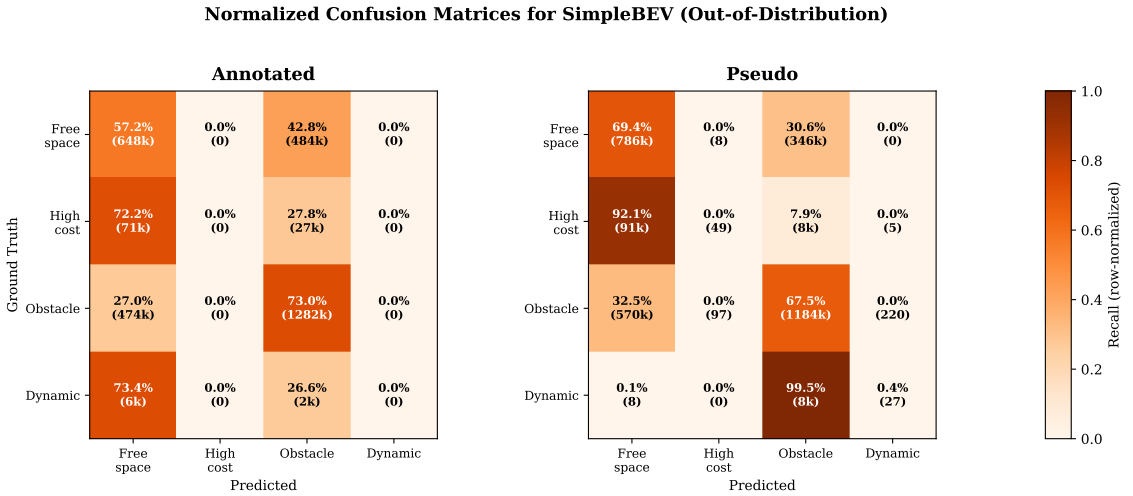


Figure 5.4: Confusion matrices for SimpleBEV on the out-of-distribution test set, comparing annotated and pseudo-label training.

Figure 5.5 provides a visual summary of the mIoU results across all models and training methods, for both in-distribution and out-of-distribution evaluation.

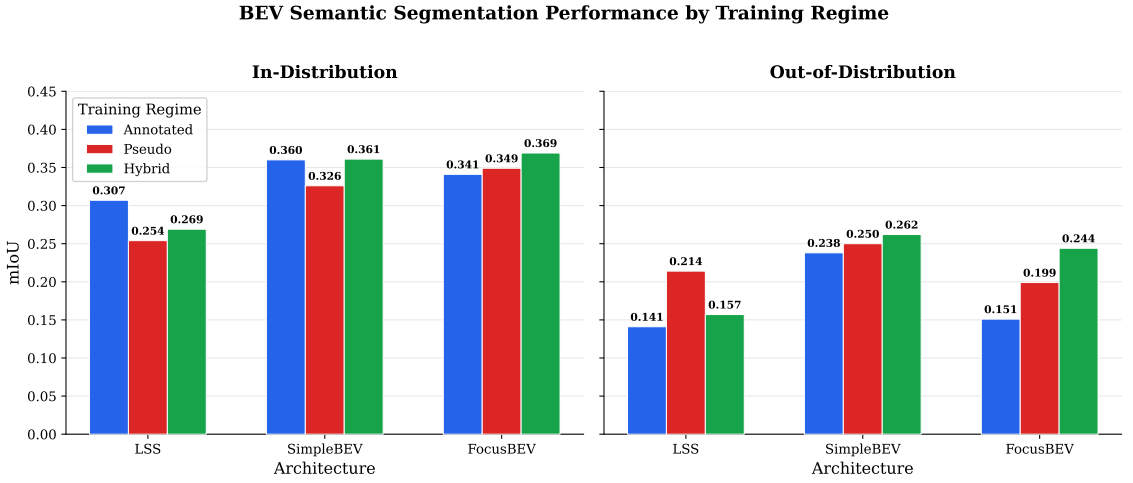


Figure 5.5: mIoU comparison across all models and training methods for OOD and ID.

Figure 5.6 displays the generalization gap by computing the percentage change in mIoU when moving from in-distribution to OOD evaluation. Across all three architectures, pseudo-label training consistently shows the smallest performance drop, supporting the hypothesis that large scale pseudo-labels can improve generalization.

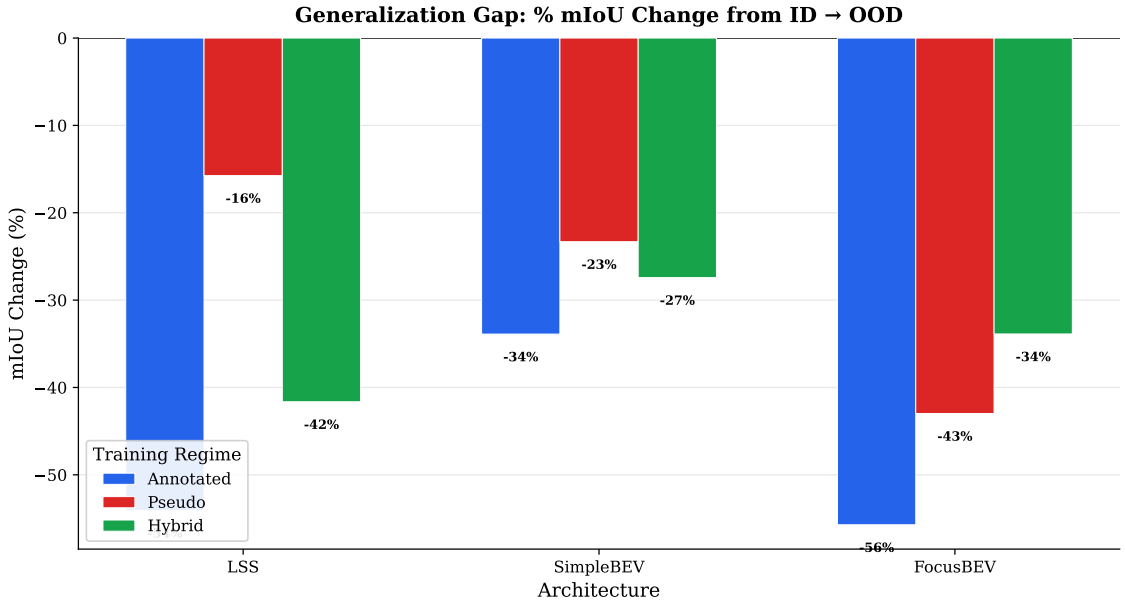


Figure 5.6: Percentage change in mIoU from in-distribution to OOD testing. Pseudo-label models (red) degrades the least amount between datasets.

5.2.5 Comparison to Baseline

Test set	Baseline	Free space	High cost	Obstacle	Dynamic	mIoU
ID	Constant (free space)	0.910	0.000	0.000	0.000	0.227
	Prior-frequency	0.835	0.000	0.047	0.000	0.220
OOD	Constant (free space)	0.378	0.000	0.000	0.000	0.095
	Prior-frequency	0.233	0.017	0.415	0.001	0.166

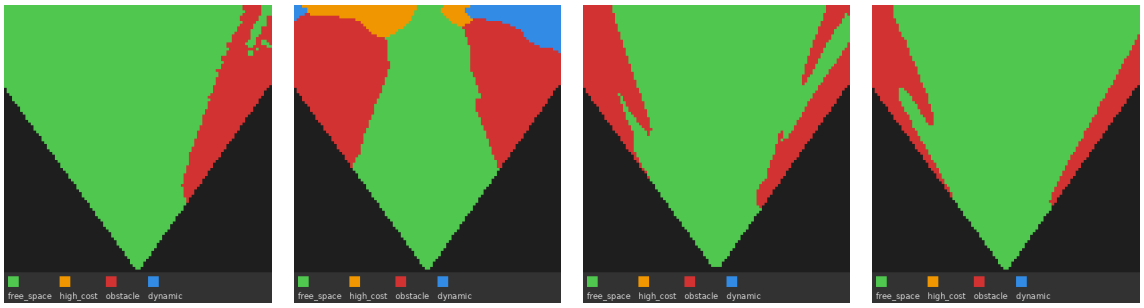
Table 5.8: Label-free reference baselines on each test set. The constant baseline predicts free space on every valid BEV cell. The prior-frequency baseline assigns classes at random in proportion to test-set frequencies, giving expected $\mathrm{IoU}_c = p_c/(2 - p_c)$. Neither uses image input.

Table 5.8 reports these baselines. In-distribution, the constant and prior-frequency baselines reach 0.227 and 0.220 mIoU, driven almost entirely by the dominant free-space class. All trained models exceed these values, but the margin varies: FocusBEV hybrid (0.369) sits well above the floor, whereas LSS trained on pseudo-labels (0.254) clears it by only about 0.03. A large part of the absolute in-distribution mIoU is therefore attributable to predicting the majority class, and comparisons between models are better read from the class-wise obstacle IoU than from mIoU alone. For OOD the relevant floor comes from the prior-frequency baseline rather than the constant one. Predicting free space everywhere reaches only 0.095 mIoU out of distribution, since obstacle rather than free space dominates the OOD dataset, so this constant baseline sits below every trained model and does little discriminating work.

The prior-frequency baseline, which reproduces the OOD class distribution without using the image, is stronger at 0.166 mIoU. Three configurations fall at or below it, LSS annotated (0.141), FocusBEV annotated (0.151) and LSS hybrid (0.157). Two of these are the annotated-only models, indicating that manual supervision in this low-annotation regime yields out-of-distribution predictions barely distinguishable from a label-free prior. Every pseudo-trained model clears both baselines (LSS 0.214, FocusBEV 0.199, SimpleBEV 0.250), as do all SimpleBEV variants. This shows the main finding: pseudo-label training lifts the models above the label-free floor that annotated training fails to clear out of distribution.

5.3 Qualitative Analysis

(a) RGB camera input. Image from "The great outdoors dataset" by Peng Jiang et al., 2025. Licensed under CC BY-NC-SA



(b) Ground truth. (c) LSS. (d) SimpleBEV. (e) FocusBEV.

Figure 5.7: Comparison of BEV semantic predictions for one representative frame: (a) RGB camera input, (b) ground truth, (c) LSS, (d) SimpleBEV, (e) FocusBEV. The colour legend is free space, high cost, obstacle, and dynamic. All models shown here were trained on pseudo-labels.

Figure 5.7 shows a comparison for one representative frame. All three models recover the main structure of the scene. Each predicts a wide central region of free space that opens up from the ego vehicle at the lower apex, bound by obstacles on both sides. This region corresponds to the dirt track and the grass beside it in the RGB input, since grass and dirt road are both grouped into the free-space class. Its triangular shape follows the ground truth and the forward field of view of the camera. The black areas in the lower corners lie outside the camera view and are not observed.

This agreement on the free-space region is the clearest shared result and is in line with the high free-space and obstacle IoU reported.

The main difference between the models and the ground truth, is the size and position of the obstacle regions rather than the free space. The ground truth places most of the obstacles on the right side, where a fallen log and trees appear in the RGB input, and leaves the left side mostly free. All three models instead predict obstacles on both sides. This is reasonable given the trees on both sides of the image, but it differs from the annotation. It is therefore not certain whether the models over-predict obstacles on the left, or whether the annotation is incomplete in this area. SimpleBEV and FocusBEV show differences in how much obstacle they place on the right side. SimpleBEV predicts a larger region there, closer to the extent shown in the ground truth, while FocusBEV predicts a less obstacle pixels. From a single frame it is not possible to say which is more accurate. For the pseudo-trained models, the obstacle IoU of these two models does not follow a fixed order. FocusBEV has the higher obstacle IoU in distribution (47.8% against 39.1% for SimpleBEV), while SimpleBEV has the higher value out of distribution (56.0% against 35.9%), as shown in Tables A.1 and A.2. The difference between the two models is therefore small and not consistent.

LSS produces the noisiest prediction of the three. It generates the largest obstacle regions, and it is the only model that predicts the minority classes in this frame, with high-cost and dynamic patches along the top of the grid. Neither class appears in the ground truth here, so these predictions are incorrect. They appear at the far range, at the top of the grid and farthest from the ego vehicle, which is where the forward-projection frustum expands and becomes least reliable. This behavior matches the failure mode described for LSS and helps to explain its lower scores. The same trend is shown in the appendix. Among the pseudo-trained models, LSS is the one that most often predicts the minority classes. On the out-of-distribution set it reaches a recall of 11.6% for high cost and 3.5% for dynamic, while SimpleBEV and FocusBEV stay close to zero for both (Table A.2). LSS also has by far the lowest obstacle precision in distribution (25.1%, against about 54% for the other two models in Table A.1), which means that a larger share of its obstacle predictions are incorrect.

In all three predictions the boundaries are block-like. This is a direct result of the 100×100 grid resolution. The shape of things like trees get corrupted because of the resolution, borders between free space and obstacle are uneven and small objects in the original image may not be present in the projection. With the exception of the incorrect LSS patches, the high-cost and dynamic classes are almost never seen in any prediction. This is consistent with the near-zero IoU of these classes, and reflects both their rarity in the training data and their absence from the scene. In the appendix, the IoU of high cost and dynamic never rises above about 2.5% for any pseudo-trained model, and is most often exactly zero (Tables A.1 and A.2).

Although Figure 5.7 shows a single frame, the same general pattern is observed across the test set. The models agree on where the vehicle can drive, and they disagree mainly on the obstacle boundaries at the edges of the view. The clearest

difference is that LSS produces noticeably more noise than the other two models, in particular at far range. The differences between SimpleBEV and FocusBEV are smaller and less consistent, and mostly concern how much obstacle is predicted near the edges of the field of view.

6

Discussion

The evaluation of the impact of pseudo-labeling will require an evaluation at each stage of the pipeline to determine if pseudo-labeling is a helpful addition when training models in this domain.

6.1 Pseudo-labeling Quality

The 2D evaluation shows a clear split. For the two dominant classes, free space and obstacle, CAT-Seg largely overlaps with the manually annotated labels after grouping. Reaching IoU scores of 0.904 and 0.873 respectively. However, for high cost and dynamic objects the performance drops by a lot, with IoU at 0.063 and 0.282 respectively. Free space and obstacles are generally quite visually distinct and are likely well represented in the underlying training data. However, high cost terrain presents a much more ambiguous group, for example the difference boundary between mud and a dirt road might not be very clear in all cases, the same goes for other natural features like the boundary between a puddle and wet grass. This is something that is debatable even among manually annotated data. Looking at the precision and recall for the high cost class (0.069 and 0.430) we can observe that CAT-Seg is likely detecting the general area of high cost ground but cannot reliably set its boundaries, instead over segmenting and producing a broad area.

This ambiguity may have been worsened by the process of collapsing the dataset's original 20 fine-grained categories into the 5 broad traversability classes used for training. Classes that are distinct for navigation may share almost identical visual textures and spectral profiles. A shallow puddle on a gravel road, for example, blends into the surrounding free space. Because the CAT-Seg makes its predictions based on 2D visual appearance rather than physical material, it struggles to disentangle high cost terrain from free space when their visual characteristics are very similar. This suggests that relying on RGB features may be insufficient for reliable high-cost terrain detection.

6.2 In Distributions Testing

On the in distribution test set the performance difference between pseduo-label and annotated training was smaller than first expected given the performance of

psedo-labaled data in chapter 5. SimpleBEV drops from 0.360 (annotated) to 0.326 (pseudo), while FocusBEV improves, going from 0.341 to 0.349 in for the pseudo labeled training. LSS shows the largest drop at 0.307 compared to 0.254.

As shown in Table 5.8, a constant free space predictor already reaches 0.227 mIoU in distribution, so the raw mIoU values in Tables 5.4–5.6 should be read against this floor, the meaningful signal lies in the obstacle IoU and the margin above baseline.

Looking at the detailed per class IoU, it can be observed that high cost and dynamic classes score zero across every model and both training sets. This is likely a result of two different problems, the pseudo-label quality for these 2 classes are not great, and the support in the BEV dataset is very small at 695 BEV cells with high cost and 156 for dynamic. Weighted cross-entropy was used as a measure to counteract the imbalance, but ultimately with these two classes being so rare it cannot compensate. This is often the reason that semantic classes are collapsed into fewer broader classes in this domain of off road terrain semantic segmentation. However, unlike the OOD test set, the class distribution in the annotated training dataset proved to be too imbalanced even with the 22 classes collapsed into fewer broad ones.

Its also important to note that the in distribution test does not provide a fair test for the pseudo-labeled trained models as the test set is compromised of annotated data from the same dataset. This means these pseudo-labeled dataset is inherently at a disadvantage. However, despite this disadvantage the models trained on the pseduo-labeled data is still largely quite competitive. Showing that the datasets large scale can make up for worse quality annotations.

6.3 Out Of Distribution Generalization

The OOD results show that pseudo-label training generalizes better than annotated training for all three architectures. FocusBEV goes from 0.151 mIoU on annotated to 0.199 on pseudo. LSS goes from 0.141 to 0.214. SimpleBEV goes from 0.238 to 0.250, a margin within the interpretability band defined in 5.2, so for this architecture pseudo-labels are best read as matching rather than beating annotated training out of distribution.

The trivial baselines shows this clearly, the prior-frequency baseline reaches 0.166 mIoU out of distribution, above the annotated LSS and FocusBEV models. That annotated training does not reliably beat a label-free baseline out of distribution, while every pseudo-trained model does, is arguably the clearest single piece of evidence for the generalization benefit of pseudo-labels.

This could be due to what each supervision source teaches the model. The annotated labels were drawn in a single environment and reflect the boundaries set by the annotators for that environment. A model trained on them might reproduce well in distribution and fails when the terrain changes. The pseudo-labels from CAT-Seg are noisier, but they transfer the semantics of the VFM that produced them, which was trained on a far wider range of scenes. As such the models trained on them possibly learn a more environment independent representation of each class. We

cannot confirm this mechanism from the present experiments, but it is consistent with the pseudo-label models degrading less than the annotated models in every case (Figure 5.6).

SimpleBEV has the strongest OOD obstacle IoU in every training condition, reaching 0.582 under hybrid training. This is worth noting because FocusBEV is the stronger model for obstacle detection in distribution. One reading is that SimpleBEV's fixed voxel sampling does not commit to a learned depth prior and so degrades less when the geometry changes, while the cycle transformation in FocusBEV fits the training-set geometry more closely and transfers less well. This is one possible explanation rather than a demonstrated one. SimpleBEV also uses a ResNet-50 backbone against the EfficientNet-B0 used by the other two models, so part of the difference may come from backbone capacity rather than from the projection method alone.

LSS under pseudo training has the highest raw obstacle IoU of any single model at 0.446, but its free space IoU is only 0.370, which pulls its overall mIoU below SimpleBEV. The pattern fits LSS over-predicting obstacle in the OOD domain, which is the expected behaviour of the forward-projection depth distribution when it meets unfamiliar terrain.

The dynamic and high cost classes stay near zero in OOD, as they did in distribution. The only values that rise clearly above zero are the high cost IoUs for LSS, around 0.025 under pseudo training and 0.027 under hybrid training. Given how little support the class has, these are more likely noise than a real signal.

It is worth setting these results against the original SimpleBEV work, which used dense radar and LiDAR and accurate metric geometry. Here the same projection idea was driven by monocular depth from Depth Anything V2 and still gave the most robust OOD performance. This is consistent with the inverse projection being more tolerant of depth noise than the forward projection in LSS, which predicts an explicit discrete depth distribution and can smear features into the wrong cells when its depth priors meet OOD terrain.

6.4 Finetuning

The tests also try a finetuning regime. The models trained on the pseudo-labeled data are then fine-tuned on the annotated data set but with a frozen backbone. This generally gave mixed results, for FocusBEV it works well, with mIoU going from 0.349 to 0.369, this gain sits at the edge of the interpretability band, so it is suggestive rather than firmly established.. For SimpleBEV the hybrid result (0.361) matches annotated only training, which means finetuning on annotations after pseudo-label pretraining costs nothing. For LSS it looks worse, with hybrid training (0.269) falling below annotated only (0.307), indicating the model has difficulty integrating the two annotation sources.

FocusBEV, however, seems to be the model which benefits the most from finetuning. Given the similarity to LSS, the results point to the cycle mechanism benefits

from pretraining on larger noisier datasets and then being refined on the precise annotations of the annotated datasets, likely tightening the semantic boundaries.

6.5 Qualitative Behaviour

The qualitative comparison in Figure 5.7 adds information that the metrics do not show. The mIoU values describe how well each model performs on average, but they do not show where in the scene the errors occur, or what kind of error each model makes. The predicted maps make this clearer.

The most notable observation is that all three models agree on the central free-space area. For an off-road vehicle, the main task of the perception system is to find the area where the vehicle can drive. The predictions show that this part of the task is solved reasonably well by every model and despite LSS poor perfomance at far away cells, even under pseudo-label training and even though the mIoU values are poor. This suggests that the modest mIoU is caused mainly by the difficult minority classes and by the exact placement of the obstacle boundaries, instead of a failure to identify drivable terrain.

The qualitative results also help to explain the differences between the models that are seen in the metrics. LSS is the only model that produces clearly visible noise, in particular at far range, where it predicts obstacles, high cost, and dynamic objects that are not present. This is consistent with the forward-projection mechanism.

At long range the depth becomes wide and uncertain, so small errors in the predicted depth distribution spread image features into the wrong grid cells. This effect is difficult to see in the mIoU value, but it is clear in the predicted map. For SimpleBEV and FocusBEV the visual differences are smaller and do not clearly favour one model, which agrees with the inconsistent ordering of these two models in the metrics.

Additional problems are likely to arise due to the method to which the BEV targets are created, by using infill it opens up to the possibility that occluded regions are not reliably labeled in BEV space. For example a region behind a larger obstacle is likely to be filled in as an obstacle as well while it realistically should not be valid. This is likely and issue most present further away in the BEV space. The reason infill was implemented was to fill smaller gaps at far ranges, which would otherwise have left streaks in BEV space. Given this tradeoff, its likely from a qualitative analysis that infill is practical and useful despite its draws. Although the set radius for the infill can probably be tweaked.

6.6 Limitations

The annotated data set is limited in size and was collected in one single environment. Furthermore, manual annotation at terrain boundaries is subjective, which is likely to influence and introduce errors between the OOD an in distribution datasets. To

mitigate this, the two datasets selected were published by the same organization with a large overlap in classes. Where The Great Outdoors contained an additional 2 classes compared to Rellis 3D which were then compacted to the 5 broad classes.

Both datasets were also initially segmented by a deep learning model called SAM which is likely to have influenced the broader selection among the 22 classes to different regions in the image and just simple refined by hand. When it is possible a human might have given the same region a different class entirely if not given a rough semantic mask by the model before. However, this is very likely mitgated by the compacting to 5 broader classes after it is applied to both datasets.

The 100×100 BEV grid at 0.25m resolution means thin structures like fence posts or narrow tree trunks may fall within a single cell alongside other terrain, making detection unreliable at any resolution. The persistent failure to detect dynamic objects across all architectures highlights this fundamental limitation of the current pipeline. The issue is twofold: severe class imbalance and spatial resolution. At a 0.25m per cell resolution, a pedestrian occupies less than a single cell and is easily smoothed out by the dominant surrounding terrain during convolutional feature pooling. Furthermore, because foundation models are often trained on internet images, they can likely segment broad natural background features well but may under-predict rare, objects in highly specific off-road contexts.

Furthermore, all three datasets are evaluated on BEV space ground truth that is generated using the same projection pipeline. This pipeline will naturally introduce noise to the ground truth BEV maps. Given the state of existing datasets within the off road terrain, this is type BEV space data is not contained in any dataset. As such it is largely unavoidable.

All results in this thesis are reported from single training runs. Due to compute and time constraints, models were not retrained across multiple random seeds, so we do not report variance or perform significance testing. Several of the observed differences, particularly the in distribution gaps between training regimes and the smaller OOD margins are modest, and we therefore treat them as indicative rather than statistically established. The larger OOD gaps are more significant to this than the smaller ones. Multi seed runs or other verifications will be left to future work. Although anecdotally it is worth it mention that throughout development we performed several training runs as new improvements where made, and with this noticed that the variance in the validation loss and test set mIoU were not very high.

6.7 Societal, Ethical and Ecological Aspects

The work in this thesis is intended for autonomous UGV navigation in off-road environments, with possible military and civilian use cases. While the work focuses on semantic BEV segmentation, deployment context raises several ethical considerations.

One big concern is the near-zero detection performance on the dynamic class since it

includes people and vehicles. Across all models and training methods tested in this project, dynamic objects are effectively invisible to the system. Providing a UGV with such a system in environments where humans or animals may be present is a big safety risk. This is driven by class imbalance in the training data rather than by the pseudo-labeling approach itself.

Another concern is the use of pseudo-labels generated by vision foundation models trained on large-scale web data. These models may carry biases from their pretraining distribution, for example underrepresenting some terrain types, lighting conditions, or objects. Biases can spread into the training targets since these models are trained mostly without human review and may later spread into the student models. The hybrid training may reduce this bias by the addition of human annotations, but the majority of the training data remains unreviewed.

The pipeline developed here is dual-use by nature. While the intended application is defensive, the same technology could enable offensive use cases if integrated into armed platforms. This concern applies to most autonomous navigation research.

Lastly, there is the cost of training deep learning models. All experiments were performed on GPU compute nodes. Pseudo-label generation pipeline processed approximately 18,000 frames through multiple foundation models. Although the footprint is small in this project, scaling self-supervised pipelines to larger datasets and heavier models will increase energy consumption. This should be weighed and compared against the human cost of manual annotation it replaces.

6.8 Future Work

The work presented in this thesis opens several possibilities for further development, both on the data side and on the modeling side.

The most direct improvement would be to raise the quality of the pseudo-labels themselves. An alternative approach would be to fine-tune the VLM on a small set of off-road images, lifting pseudo-label quality for domain-specific categories without requiring full manual annotation of the training set. Rather than annotating uniformly, an active learning strategy could further focus human effort on the frames or image regions where pseudo-labels are most uncertain, where it is most impactful.

The near-zero performance on high cost and dynamic classes is fundamentally a data problem. Targeted data collection campaigns focusing on these underrepresented categories, combined with augmentation techniques, could make them learnable. Alternative loss functions may also help where weighted cross-entropy proved insufficient. On the architecture side, the current 100×100 grid at 0.25 m resolution limits the ability to detect small or thin structures. A multi-scale approach, similar to the one used in RoadRunner M&M, could maintain fine-grained resolution near the vehicle while providing better coverage at range. Additionally, the pipeline currently treats each frame independently. Temporal consistency across consecutive frames could smooth noisy predictions and improve recall for transient dynamic objects.

Finally, the generalization results raise questions that deserve deeper investigation. This thesis showed that pseudo-label trained models can outperform manually supervised ones out of distribution, but the evaluation is limited to a single OOD dataset. Testing on additional off-road datasets from different geographic regions and terrain types would strengthen the generalization claims. A particularly interesting follow-up would be to study region-specific fine-tuning. If a pseudo-label model is fine-tuned on data from a specific deployment region, does it keep its OOD advantage or does it collapse back to the domain-specific behavior observed with annotated-only training? Understanding this trade-off is very relevant for UGVs that are expected to operate across multiple and potentially unfamiliar terrain types.

7

Conclusion

This thesis investigated whether pseudo-labeled training data generated automatically using Vision Foundation Models could realistically replace or reduce dependence on manual annotation for semantic BEV perception in off-road environments.

The first research question aimed to compare the performance of pseudo-label training against manual supervision. On the in-distribution test set, pseudo-label trained models perform similar to the models trained on manual data. SimpleBEV drops only from 0.360 on manual to 0.326 on pseudo. FocusBEV performs comparably under the two regimes (0.341 vs. 0.349), a difference within the interpretability band.. High cost and dynamic classes score zero across all models and training conditions, a result driven by extreme class imbalance rather than by the label source. The hybrid training strategy performed similar or better than the annotated-only baseline for all architectures, with FocusBEV achieving the best overall mIoU of 0.369. This indicates that pseudo-labels can be effective pre-training data and that quality can be improved by a small amount of supervised fine-tuning.

The second research question evaluated the quality of the zero-shot pseudo-labels themselves. The thesis showed that CAT-Seg produces usable pseudo-labels for the classes that matter most, with free space and obstacles reaching IoU scores of 0.904 and 0.873 against manual annotations in 2D. High cost terrain and dynamic objects showed poor results in this step too, with IoU of 0.063 and 0.282 respectively. Precision-recall analysis revealed that the model finds the general area but not precise enough boundaries for high cost. This led to over-segmentation. Both classes are so underrepresented in the dataset that even heavily reweighted loss functions hardly impact training. This shows that while VLMs can capture broad boundaries, the fine semantic borders in ambiguous terrain remain a data collection challenge.

The final research question investigated out-of-distribution generalization. The most notable finding of this thesis is that pseudo-label trained models outperformed manually supervised ones on the specific OOD test set used in this project. The relative improvement ranged from 5% to nearly 50%, essentially inverting the in-distribution ranking. Manual annotations in these low annotated regimes appear to teach models to reproduce dataset-specific labeling. These do not transfer well to new environments. Labels created by VLMs however, carry semantics from broad pretraining, which seem to generalize better across terrain domains. The generalization gap analysis shows that pseudo-label models had the smallest mIoU decline when comparing in-distribution to OOD evaluation across all three architectures.

Among the architectures evaluated to answer these questions, SimpleBEV was the most consistent, leading on out-of-distribution performance across all training conditions in our experiments. FocusBEV performed best in-distribution, particularly on obstacle detection, and benefited most from hybrid training. LSS was the weakest of the three in most conditions in our experiments.

These results show that a perception pipeline built on passive sensors, with offline geometric LiDAR supervision, can produce BEV models that transfer to new environments without any manual annotation. For a field where datasets are limited, often kept private, and deployment conditions rarely match training ones, that is a meaningful result.

Bibliography

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

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

[4] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: A survey. *CoRR*, abs/2202.02980, 2022.

[5] Peng Jiang, Philip R. Osteen, Maggie B. Wigness, and Srikanth Saripalli. RELLIS-3D dataset: Data, benchmarks and analysis. *CoRR*, abs/2011.12954, 2020.

[6] Peng Jiang, Kasi Viswanath, Akhil Nagariya, George Chustz, Maggie Wigness, Philip Osteen, Timothy Overbye, Christian Ellis, Long Quang, and Srikanth Saripalli. Go: The great outdoors multimodal dataset, 2025.

[7] Samuel Triest, Matthew Sivaprakasam, Sean J. Wang, Wenshan Wang, Aaron M. Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models, 2022.

[8] Matthew Sivaprakasam, Parv Maheshwari, Mateo Guaman Castro, Samuel Triest, Micah Nye, Steve Willits, Andrew Saba, Wenshan Wang, and Sebastian Scherer. Tartandrive 2.0: More modalities and better infrastructure to further self-supervised learning research in off-road driving tasks, 2024.

[9] Nuanchen Lin, Wenfeng Zhao, Shenghao Liang, and Minyue Zhong. Real-time segmentation of unstructured environments by combining domain generalization and attention mechanisms. *Sensors*, 23(13), 2023.

[10] Javier Gibran Apud Baca, Thomas Jantos, Mario Theuermann, Mohamed Amin Hamdad, Jan Steinbrener, Stephan Weiss, Alexander Almer, and Roland Perko. Automated data annotation for 6-dof ai-based navigation algorithm development. *Journal of Imaging*, 7(11), 2021.

[11] Xing Chen, Yujiao Dong, Xinyong Li, Xunjia Zheng, Hui Liu, and Tengyuan Li. Drivable area recognition on unstructured roads for autonomous vehicles using an optimized bilateral neural network. *Scientific Reports*, 15, 04 2025.

[12] Jouni Rantakokko and Jonas Nygårds. Obemannade farkoster för markstriden: Erfarenheter från ukraina. Technical Report FOI-R–5723–SE, Totalförsvarets forskningsinstitut (FOI), August 2025.

[13] Yafeng Bu, Zhenping Sun, Xiaohui Li, Jun Zeng, Xin Zhang, and Hui Shen. Self-supervised traversability learning with online prototype adaptation for off-road autonomous driving. *IEEE Robotics and Automation Letters*, 10(11):11259–11266, 2025.

[14] Manthan Patel, Jonas Frey, Deegan Atha, Patrick Spieler, Marco Hutter, and Shehryar Khattak. Roadrunner mm – learning multi-range multi-resolution traversability maps for autonomous off-road navigation, 2024.

[15] Kasi Viswanath, Kartikeya Singh, Peng Jiang, P. B. Sujit, and Srikanth Saripalli. Offseg: A semantic segmentation framework for off-road driving. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 1550–1555. IEEE, 2021.

[16] Anurag Das, Yongqin Xian, Yang He, Zeynep Akata, and Bernt Schiele. Urban scene semantic segmentation with low-cost coarse annotation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5978–5987, 2023.

[17] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning, 2020.

[18] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[19] Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, JoonHo Lee, Wentao Yuan, Zoey Chen, Samuel Deng, Greg Okopal, Dieter Fox, Byron Boots, and Amirreza Shaban. Terrainnet: Visual modeling of complex terrain for high-speed, off-road navigation, 2023.

[20] Aniket Datar, Anuj Pokhrel, Mohammad Nazeri, Madhan B. Rao, Chenhui Pan, Yufan Zhang, Andre Harrison, Maggie Wigness, Philip R. Osteen, Jinwei Ye, and Xuesu Xiao. M2p2: A multi-modal passive perception dataset for off-road mobility in extreme low-light conditions, 2025.

[21] Kasi Viswanath, P. B. Sujit, and Srikanth Saripalli. Camel: Learning cost-maps made easy for off-road driving. In *Machine Learning for Autonomous Driving Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, New Orleans, USA, 2022. Workshop paper.

[22] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. *CoRR*, abs/2008.05711, 2020.

[23] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception?, 2022.

[24] Jiawei Zhao, Qixing Jiang, Xuede Li, and Junfeng Luo. Focus on bev: Self-calibrated cycle view transformation for monocular birds-eye-view segmentation, 2024.

[25] Adam Lilja, Ji Lan, Junsheng Fu, and Lars Hammarstrand. Queryocc: Query-based self-supervision for 3d semantic occupancy. *arXiv preprint arXiv:2511.17221*, 2025.

[26] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2nd edition, 2003.

[27] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *CoRR*, abs/1811.08188, 2018.

[28] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[30] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

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

[32] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *CoRR*, abs/2112.01527, 2021.

[33] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus,

Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren E. Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021.

[34] Junsu Kim, Yunhoe Ku, Jihyeon Kim, Junuk Cha, and Seungryul Baek. Vlm-pl: Advanced pseudo labeling approach for class incremental object detection via vision-language model, 2024.

[35] Timo Lüddecke and Alexander S. Ecker. Image segmentation using text and image prompts, 2022.

[36] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation, 2024.

[37] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[38] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[39] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[40] Maggie Wigness, Sungmin Eum, John G. Rogers, David Han, and Heesung Kwon. A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5000–5007, 2019.

[41] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *CoRR*, abs/1710.05381, 2017.

[42] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[43] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.

A

Appendix 1

Model	Data	Class	Support (GT)	Recall	Precision	F1-Score	IoU
LSS	Annotated	free_space	1,068,112	97.31%	94.13%	95.70%	91.75%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	39.05%	59.73%	47.23%	30.91%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Pseudo	free_space	1,068,112	87.00%	93.98%	90.36%	82.41%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	44.08%	25.12%	32.00%	19.05%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Hybrid	free_space	1,068,112	90.39%	93.87%	92.10%	85.35%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	38.84%	34.17%	36.35%	22.22%
		dynamic	156	12.18%	0.07%	0.15%	0.07%
SimpleBEV	Annotated	free_space	1,068,112	96.40%	96.86%	96.63%	93.49%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	68.99%	65.32%	67.10%	50.49%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Pseudo	free_space	1,068,112	95.14%	95.81%	95.47%	91.33%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	58.41%	54.12%	56.19%	39.07%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Hybrid	free_space	1,068,112	96.46%	96.87%	96.66%	93.54%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	68.58%	66.20%	67.37%	50.79%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
FocusBEV	Annotated	free_space	1,068,112	98.05%	95.33%	96.67%	93.55%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	51.45%	72.21%	60.09%	42.94%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Pseudo	free_space	1,068,112	93.74%	97.69%	95.68%	91.71%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	78.22%	55.14%	64.68%	47.80%
		dynamic	156	0.00%	0.00%	0.00%	0.00%
	Hybrid	free_space	1,068,112	97.49%	96.68%	97.08%	94.33%
		high_cost	695	0.00%	0.00%	0.00%	0.00%
		obstacle	104,999	66.65%	72.36%	69.39%	53.13%
		dynamic	156	0.00%	0.00%	0.00%	0.00%

Table A.1: In-Distribution Performance (Tested on Annotated Data)

Model	Training Data	Class	Support (GT)	Recall	Precision	F1-Score	IoU
LSS	Annotated	free_space	1,132,548	98.98%	40.21%	57.19%	40.04%
		high_cost	98,362	0.00%	0.00%	0.00%	0.00%
		obstacle	1,755,170	10.56%	95.07%	19.01%	10.51%
		dynamic	7,675	13.33%	9.68%	11.22%	5.94%
	Pseudo	free_space	1,132,548	60.04%	49.11%	54.03%	37.01%
		high_cost	98,362	11.64%	3.02%	4.80%	2.46%
		obstacle	1,755,170	52.31%	75.19%	61.70%	44.61%
		dynamic	7,675	3.50%	2.91%	3.18%	1.61%
	Hybrid	free_space	1,132,548	57.53%	48.52%	52.64%	35.72%
		high_cost	98,362	28.02%	2.92%	5.29%	2.71%
		obstacle	1,755,170	24.44%	72.65%	36.57%	22.38%
		dynamic	7,675	28.76%	1.90%	3.57%	1.81%
SimpleBEV	Annotated	free_space	1,132,548	57.24%	54.09%	55.62%	38.52%
		high_cost	98,362	0.00%	0.00%	0.00%	0.00%
		obstacle	1,755,170	73.02%	71.39%	72.20%	56.49%
		dynamic	7,675	0.00%	0.00%	0.00%	0.00%
	Pseudo	free_space	1,132,548	69.41%	54.32%	60.94%	43.83%
		high_cost	98,362	0.05%	30.49%	0.10%	0.05%
		obstacle	1,755,170	67.48%	76.60%	71.75%	55.95%
		dynamic	7,675	0.35%	10.34%	0.68%	0.34%
	Hybrid	free_space	1,132,548	64.81%	56.06%	60.11%	42.97%
		high_cost	98,362	0.00%	0.00%	0.00%	0.00%
		obstacle	1,755,170	72.04%	75.23%	73.60%	58.23%
		dynamic	7,675	4.52%	14.45%	6.89%	3.57%
FocusBEV	Annotated	free_space	1,132,548	98.54%	42.14%	59.04%	41.88%
		high_cost	98,362	0.00%	0.00%	0.00%	0.00%
		obstacle	1,755,170	18.53%	94.99%	31.01%	18.35%
		dynamic	7,675	0.00%	0.00%	0.00%	0.00%
	Pseudo	free_space	1,132,548	88.35%	46.05%	60.54%	43.41%
		high_cost	98,362	0.00%	0.00%	0.00%	0.00%
		obstacle	1,755,170	38.72%	83.04%	52.82%	35.88%
		dynamic	7,675	0.47%	3.06%	0.81%	0.41%
	Hybrid	free_space	1,132,548	85.86%	52.51%	65.17%	48.33%
		high_cost	98,362	0.05%	0.16%	0.08%	0.04%
		obstacle	1,755,170	53.72%	85.01%	65.84%	49.07%
		dynamic	7,675	0.04%	0.21%	0.07%	0.03%

Table A.2: Out-of-Distribution Performance (Tested on OOD Data).

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY