SkyEye: Self-Supervised Bird's-Eye-View Semantic Mapping Using Monocular Frontal View Images

Nikhil Gosala[1]* Kürsat Petek[1]* Paulo L. J. Drews-Jr[1,2] Wolfram Burgard[2] Abhinav Valada[1]
[1]University of Freiburg [2]Federal University of Rio Grande [3]University of Technology Nuremberg
http://skyeye.cs.uni-freiburg.de

Abstract

Bird's-Eye-View (BEV) semantic maps have become an essential component of automated driving pipelines due to the rich representation they provide for decision-making tasks. However, existing approaches for generating these maps still follow a fully supervised training paradigm and hence rely on large amounts of annotated BEV data. In this work, we address this limitation by proposing the first self-supervised approach for generating a BEV semantic map using a single monocular image from the frontal view (FV). During training, we overcome the need for BEV ground truth annotations by leveraging the more easily available FV semantic annotations of video sequences. Thus, we propose the SkyEye architecture that learns based on two modes of self-supervision, namely, implicit supervision and explicit supervision. Implicit supervision trains the model by enforcing spatial consistency of the scene over time based on FV semantic sequences, while explicit supervision exploits BEV pseudolabels generated from FV semantic annotations and self-supervised depth estimates. Extensive evaluations on the KITTI-360 dataset demonstrate that our self-supervised approach performs on par with the state-of-the-art fully supervised methods and achieves competitive results using only 1% of direct supervision in BEV compared to fully supervised approaches. Finally, we publicly release both our code and the BEV datasets generated from the KITTI-360 and Waymo datasets.

1. Introduction

Bird's-Eye-View (BEV) maps are an integral part of an autonomous driving pipeline as they allow the vehicle to perceive the environment using a feature-rich yet computationally-efficient representation. These maps capture both static and dynamic obstacles in the scene while encoding their absolute distances in the metric scale using a low-cost 2D representation. Such characteristics allow them to be used in many distance-based time-sensitive applications such as trajectory estimation and collision avoidance [12, 14]. Existing approaches that estimate BEV maps from frontal view (FV) images and/or LiDAR scans require large datasets annotated in the BEV as they are trained in a fully supervised manner [6, 19, 23, 43]. However, BEV ground truth generation relies on the presence of HD maps, annotated 3D point clouds, and/or 3D bounding boxes, which are extremely arduous to obtain [27]. Recent approaches [29, 36] circumvent this problem of requiring BEV ground truths by leveraging data from simulation environments. However, these approaches suffer from the large domain gap between simulated and real-world images, which results in their reduced performance in the real world.

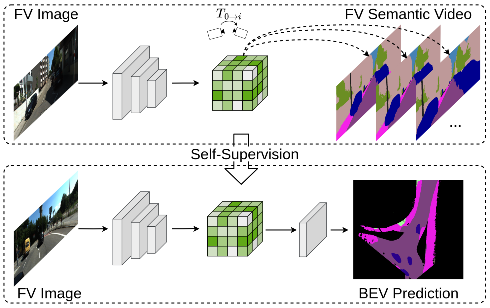

Figure 1. SkyEye: The first self-supervised framework for semantic BEV mapping. We use sequences of FV semantic annotations to train the network to estimate a semantic map in BEV using a single RGB input.

In this work, we address the aforementioned limitations by proposing *SkyEye*, the first self-supervised learning framework for generating an instantaneous semantic map in BEV, given a single monocular FV image. During training, our approach, depicted in Fig. 1, overcomes the need for BEV ground truths by leveraging FV semantic ground truth labels along with the spatial and temporal consistency offered by video sequences. FV semantic ground truth labels can easily be obtained with reduced human annotation effort due to the relatively small domain gap between FV images of different datasets which allows for efficient label transfer [15, 17, 37]. Additionally, no range sensor is required for data recording.

*Equal contribution

During inference, our model only uses a single monocular FV image to generate the semantic map in BEV.

Our proposed self-supervised learning framework leverages two supervision signals, namely, *implicit* and *explicit supervision*. Implicit supervision generates the training signal by enforcing spatial and temporal consistency of the scene. To this end, our model generates the FV semantic predictions for the current and future time steps using the FV image of only the current time step. These predictions are supervised using the corresponding ground truth labels in FV. Explicit supervision, in contrast, supervises the network using BEV semantic pseudolabels generated from FV semantic ground truths using a self-supervised depth estimation network augmented with a dedicated post-processing procedure. We perform extensive evaluations of *SkyEye* on the KITTI-360 dataset and demonstrate its generalizability on the Waymo dataset. Results demonstrate that *SkyEye* performs on par with the state-of-the-art fully-supervised approaches and achieves competitive performance with only 1% of pseudolabels in BEV. Further, we outperform all baseline methods w.r.t. generalization capabilities.

Our main contributions can thus be stated as follows:

- The first self-supervised framework for generating semantic BEV maps from monocular FV images.
- An implicit supervision strategy that leverages semantic annotations in FV to encode semantic and spatial information into a latent voxel grid.
- A pseudolabel generation pipeline to create BEV pseudolabels from FV semantic ground truth labels.
- A novel semantic BEV dataset derived from Waymo.
- Extensive evaluations as well as ablation studies to show the impact of our contributions.
- Publicly available code for our *SkyEye* framework at http://skyeye.cs.uni-freiburg.de.

2. Related Work

In this section, we review the existing work related to BEV semantic mapping and self-supervised 3D representation learning based on monocular images.

BEV Mapping: BEV map generation typically involves three stages: (i) FV feature extraction using an image encoder, (ii) feature transformation from FV to BEV, and (iii) BEV map generation using the transformed features - with most approaches focusing on FV-BEV transformation. The earliest approaches, VED [24] and VPN [30] learn the FV-BEV mapping using a variational encoder-decoder architecture and a two-layer multi-layer perceptron respectively. However, they do not account for the geometry of the scene which results in their poor performance in the real world. Later approaches address this limitation by integrating scene geometry into the network design. PON [32] proposes an end-to-end network wherein a dense transformer module learns the mapping between a column in the FV image and a ray in the BEV prediction. LSS [31] uses a learnable categorical depth distribution to "lift" the FV features into the 3D space. Both these approaches, however, do not generalize across different semantic classes. PanopticBEV [6] addresses these limitations by employing a dual-transformer approach to independently map the *vertical* and *flat* regions in the scene from FV to BEV. Recently, multiple approaches [2, 33, 44] have proposed using vision transformer-based architectures to learn the FV-BEV mapping, while others have explored incorporating range sensors such as LiDARs and Radars into the BEV map generation pipeline [19, 23]. It is important to note that all the aforementioned approaches follow a fully-supervised training strategy and hence rely on BEV ground truth labels for training. Although such approaches result in state-of-the-art performance, their reliance on BEV ground truth labels severely impacts their scalability. In this paper, we propose the first self-supervised approach using only FV image sequences, their corresponding FV semantic annotations, and the ego-poses for training.

Self-Supervised Monocular 3D Representation Learning: This task forms one of the fundamental challenges of computer vision and is used in tasks such as novel view synthesis and 3D reconstruction [1]. Early works use geometry-based approaches such as structure-from-motion [34], multi-view stereo [4], and multi-hypothesis labeling [11], while recent approaches typically employ deep learning-based solutions [16, 40] to address this challenge. More recently, self-supervised approaches have been proposed to alleviate the amount of annotated data required to learn the 3D structure. Video Autoencoder [18] uses an autoencoder to learn the 3D structure of a static scene for the task of novel view synthesis. In the context of robotics, self-supervised representation learning has been used for tasks such as depth estimation [5, 8], surface normal estimation [7], optical flow [21, 22], visual-inertial odometry [9], keypoints estimation [41], stereo matching [42], image enhancement [26], and scene flow [13] among many others. These approaches have shown tremendous potential in the real world due to their ability to efficiently scale across multiple locations without needing expensive human intervention. We extend this set of self-supervised approaches by proposing the first self-supervised learning framework to predict BEV semantic maps without the need for any BEV ground truth data.

3. Technical Approach

In this section, we present our novel self-supervised learning framework, SkyEye, for generating BEV semantic maps from a single monocular FV image without any ground truth supervision in BEV. The core idea of our approach is to generate an intermediate 3D voxel grid that serves as a joint feature representation for both FV and BEV segmentation

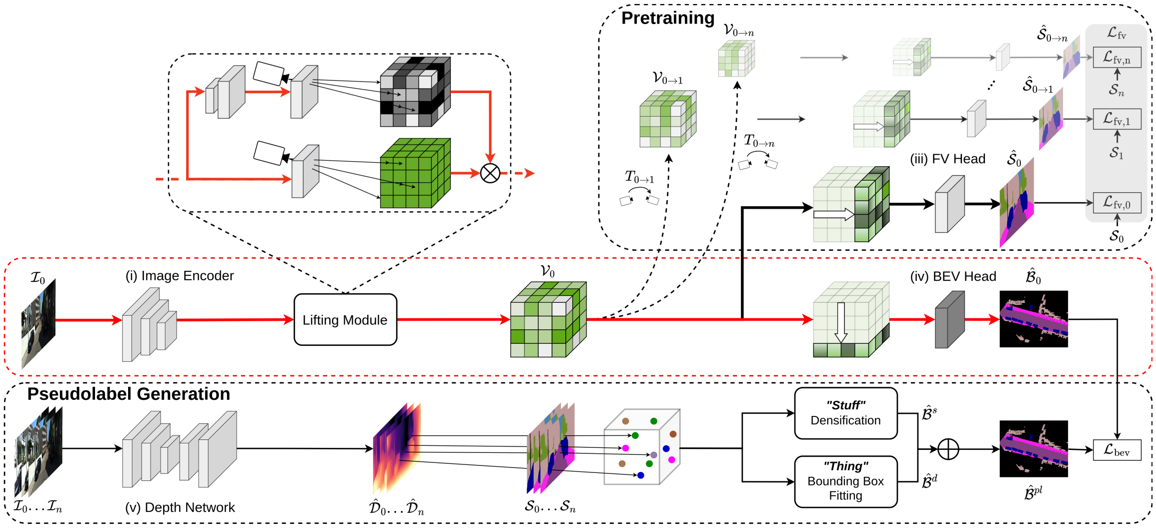

Figure 2. Overview of our proposed self-supervised BEV semantic mapping framework, SkyEye. The core component of our approach is the latent voxel grid $\mathcal{V}_0$ that serves as a joint feature representation for segmentation tasks in FV and BEV. We encode spatial and semantic information into the voxel grid using *implicit supervision* during a pretraining step and *explicit supervision* in a subsequent refinement step using pseudolabels that are generated with a self-supervised depth prediction pipeline. The path in red denotes the processing steps during inference time.

tasks, thus allowing us to leverage FV supervision to augment the BEV semantic learning procedure. An overview of our proposed self-supervised pipeline is depicted in Fig. 2.

Our framework generates the supervision signal using two strategies, namely, *implicit supervision* and *explicit supervision*. Implicit supervision generates the training signal by exploiting the spatial and temporal consistency of the scene via FV Semantic Scene Consistency ($\mathcal{L}_{fv}$, Sec. 3.2) which operates on the depth-wise projection of the voxel grid. The explicit supervision, in turn, operates on the orthographic height-wise projection of the voxel grid and provides supervision in BEV via pseudolabels generated in a self-supervised manner ($\mathcal{L}_{bev}$, Sec. 3.3). The final loss is thus computed as:

$$\mathcal{L} = \mathcal{L}_{fv} + \mathcal{L}_{bev} \tag{1}$$

In the following sections, we present an overview of our network architecture and provide further insight into the computation of the aforementioned losses. Further, in all the upcoming notations, the subscript i refers to an instance of an element at time step t_i.

3.1. Network Architecture

Our model comprises five major components: (i) an image encoder to generate 2D image features, (ii) a lifting module to generate the 3D voxel grid using a learned depth distribution, (iii) an FV semantic head to generate the FV semantic predictions for implicit supervision, (iv) a BEV semantic head to generate the BEV semantic map, and (v) an independent self-supervised depth network to generate the BEV pseudolabels. Fig. 2 presents an overview of our proposed framework.
The encoder follows the EfficientDet-D3 backbone [39] which takes an FV image as input and outputs 2D features at four different scales which we subsequently merge using the multi-scale fusion strategy outlined in EfficientPS [28]. The lifting module projects the 2D features to a 3D voxel grid representation using the camera projection equation coupled with a learned depth distribution that provides the likelihood of features in a given voxel. We then process the voxel grid depth-wise or height-wise based on whether the output is in the FV or BEV respectively. We generate the FV semantic logits by applying perspective distortion to the 3D voxel grid, flattening it along the depth dimension, and mapping it to the output channels using a 1×1 convolution. Similarly, we generate the BEV logits by flattening the 3D voxel grid orthographically along the height dimension and passing it through a 1×1 convolution to generate the required output channels in the BEV. Our self-supervised depth network is independent from the aforementioned model and is only used to generate the BEV semantic pseudolabels. It uses a separate instance of EfficientDet-D3 backbone and feature merging module outlined above. The depth decoder consists of three upsampling layers, each of which follow the upsampling strategy defined in [5]. The final depth is then computed by applying a 3×3 convolution, normalizing it using a sigmoid function and scaling it to the required range.

3.2. Implicit Supervision

Autonomous driving scenes comprise many static elements such as parked cars and buildings which establish a strong framework for generating a supervision signal by exploiting their consistency over multiple time steps. We exploit this characteristic of the real world and generate the implicit supervision signal by enforcing consistency between

FV semantic predictions at multiple time steps. To this end, we predict the FV semantic maps for the initial (t_0) as well as future time steps ($t_1, ..., t_n$) using only the intermediate voxel grid representation at the initial time step as depicted in Fig. 3. We hypothesize that this formulation would help the network generate a spatially consistent volumetric representation of the scene from a single FV image. Further, we also hypothesize that this formulation would help the voxel grid encode complementary information from multiple images to resolve occlusions and hence play a pivotal role in generating accurate BEV semantic maps from the limited view of only a single time step.

We first use the provided FV monocular image $\mathcal{I}_0$ to generate the intermediate 3D voxel grid representation $\mathcal{V}_0$. This voxel grid is perspectively distorted using the camera intrinsics and processed along the depth dimension to generate the FV semantic prediction $\hat{\mathcal{S}}_0$. Perspective distortion of the voxel grid prior to processing it in FV is crucial to prevent implicit supervision from distorting the voxel grid. Parallelly, we transform $\mathcal{V}_0$ to generate the voxel grids $\mathcal{V}_{0\to i} = T_{0\to i}\mathcal{V}_0$ for future time steps $(t_1, ...t_n)$ using the relative ego poses $T_{0\to i}$ between the initial and future time steps. We then use the generated pseudo voxel grids to infer the future FV semantic predictions $\hat{\mathcal{S}}_{0\to 1}, \hat{\mathcal{S}}_{0\to 2}, ..., \hat{\mathcal{S}}_{0\to n}$. Subsequently, we compute the cross entropy loss between the FV semantic predictions and their corresponding FV semantic ground truths to generate the implicit supervision signal for training the model. We linearly down-weight the loss for future time steps to negate the ill effects of dynamic objects and error propagation during model training. Thus, we compute the FV semantic scene consistency loss $\mathcal{L}_{fv}$ by accumulating the losses for each time step $\mathcal{L}_{fv,i}$ as:

$$\mathcal{L}_{fv} = \sum_{i=0}^{n} L_{fv,i} = \sum_{i=0}^{n} w_i CE(\hat{\mathcal{S}}_{0\to i}, \mathcal{S}_i), \qquad (2)$$

where w_i refers to the time step-based weight which linearly decays from 1 to 0.2, and $CE(a, b)$ refers to the cross entropy loss between tensors a and b.

3.3. Explicit Supervision

Our model comprises a BEV segmentation head with learnable parameters that is designed to generate the desired BEV semantic map. However, implicit supervision does not generate a gradient flow through the BEV head, which underlines the need for *explicit supervision* in BEV. To this end, we propose a pseudolabel generation procedure consisting three steps as depicted in Fig. 2, namely (i) a depth prediction pipeline to lift FV semantic annotations into BEV yielding a semantic point cloud, (ii) an instance generation module based on DBSCAN [35], and (iii) a densification module to generate dense segmentation masks from sparse depth predictions for static classes. Prior to pseudolabel generation, we train our depth network on the corresponding dataset in a self-supervised manner as proposed in [5] using the ego poses to ensure metric scale of the depth estimates.

Pseudolabel Generation: We generate pseudolabels for time step t_0 by employing a sequence $\mathcal{W}$ of FV images and their corresponding semantic ground truths. Fig. 4 illustrates the proposed pipeline. We first predict the depth map $\hat{\mathcal{D}}_i$ for each FV image $\mathcal{I}_i \in \mathcal{W}$ to lift FV semantic ground truths into BEV using the known camera intrinsics and poses. We then accumulate the semantic point clouds and transform them into the perspective of t_0, to obtain a single accumulated semantic point cloud $\dot{\mathcal{P}}_0 = \bigcup_{k\in\mathcal{W}} \hat{\mathcal{P}}_{k\to 0}$. For dynamic objects, we retain only those points in the point cloud that are consistent with the FV semantic ground truth at t_0 to prevent object motion from corrupting the supervision signal. We then use the accumulated point cloud to both create dense semantic labels for static classes and fit boxes around dynamic objects (Fig. 2).
First, we orthographically project points belonging to static classes to generate a sparse BEV map. We then densify this map by applying a series of morphological dilate and erode operations to generate the first set of labels $\hat{\mathcal{B}}^s$ for static classes. Second, we try to mitigate the lack of observability of the shape of dynamic objects by fitting boxes around each object instance. However, we are faced with two challenges: (i) the FV data has no notion of object instances, and (ii) the predicted depth maps are prone to outliers due to transparent and reflective surfaces. We address these challenges by introducing the notion of instances and rejecting outliers in depth maps. We do so by clustering the accumulated semantic point cloud using DBSCAN to yield a set $\mathcal{M}$ of C clusters, where $\mathcal{M} = \{m_j, j = 1...C\}$. We then project the M clusters orthographically into BEV and fit an ellipse $\mathcal{E} = \{x_c, y_c, a, b, \theta\}$, which serves as a differentiable replacement of a bounding box, around each cluster using the RANSAC [3] algorithm. The predicted ellipse parameters with its 2D center point (x_c, y_c), semi-minor axes (a, b) and orientation θ define the position, extents, and orientation of the bounding box in BEV, respectively. This procedure generates a second BEV map $\hat{\mathcal{B}}^d$ which is then overlaid on $\hat{\mathcal{B}}^s$ to generate the final pseudolabel map $\hat{\mathcal{B}}^{pl}$ containing both static and dynamic classes. Finally, this pseudolabel map is used to further supervise the semantic BEV map at the network output using the cross-entropy loss as

$$\mathcal{L}_{bev} = CE(\hat{\mathcal{B}}^{pl}, \hat{\mathcal{B}}). \qquad (3)$$

4. Experimental Results

In this section, we present the quantitative and qualitative results of our proposed self-supervised BEV semantic map generation pipeline, SkyEye, along with comprehensive ablation studies to highlight the importance of our contributions. We also present the datasets used for experimental evaluation

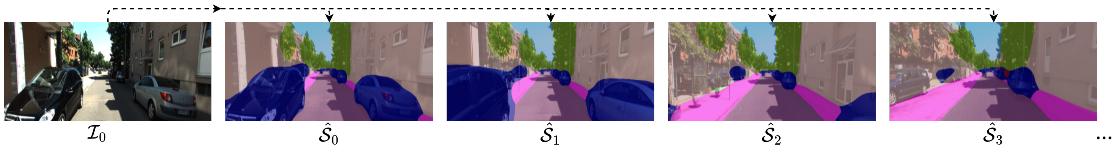

Figure 3. Semantic predictions of SkyEye for future time steps using the FV image of only the initial time step. The disocclusion of sidewalks in the semantic predictions indicates that SkyEye can reason about both occluded regions and spatial extents of objects in the scene with the encoded semantic information.

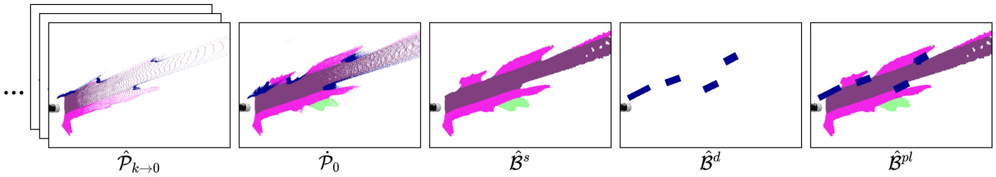

Figure 4. Overview of our pseudolabel generation pipeline. We lift semantic annotations in FV into the 3D world ($\hat{\mathcal{P}}_{k\rightarrow0}$) and accumulate them ($\dot{\mathcal{P}}_0$). We then densify the static classes in BEV ($\hat{\mathcal{B}}^s$) and fit boxes around clustered dynamic objects ($\hat{\mathcal{B}}^d$). $\hat{\mathcal{B}}^s$ and $\hat{\mathcal{B}}^d$ are merged to generate BEV pseudolabels $\hat{\mathcal{B}}^{pl}$.

and provide a detailed description of the training protocol to ensure transparency and result reproduction.

4.1. Datasets

We evaluate SkyEye on the KITTI-360 [20] dataset and study its generalization ability by pretraining it on KITTI-360 and evaluating it on Waymo Open Dataset [38]. We select these datasets to evaluate our approach on a wide variety of driving scenarios encountered in different regions of the world. Since neither KITTI-360 nor Waymo provide BEV semantic labels, we follow the data generation process outlined in PanopticBEV [6] to generate the BEV semantic ground truth labels. We slightly modify this process and remove the occlusion masking step to make BEV labels occlusion-agnostic. It is important to note that the generated BEV ground truths are only used to train the fully-supervised baselines and perform the quantitative evaluation, and are *not* used in our self-supervised learning framework. Of the 10 sequences in the KITTI-360 dataset, we use sequence 10 for validation and use the remaining sequences for training. For the generalization experiment, we evaluate our pretrained model on all samples in the Waymo validation split.

4.2. Training Protocol

We train SkyEye on images of size 1408×384 pixels by following a two-step training protocol. First, we learn to infer the 3D geometry of the scene from a single FV image by training the model using only implicit supervision on a window size of 10 for 20 epochs with a learning rate (LR) of 0.005. We sample every second image from the window to capture a long time horizon while reducing the training time of the model. We then specialize the model for BEV segmentation by explicitly supervising it using the generated BEV pseudolabels for 20 epochs and LR of 0.001. We augment the dataset during both stages using random combinations of horizontal image flipping, as well as color perturbations via changes to image brightness, contrast, and saturation. We optimize the network across both training steps using SGD with a batch size of 12, momentum of 0.9, and weight decay of 0.0001. We follow a multi-step training schedule wherein we decay LR by a factor of 0.5 at epoch 15 and 0.2 at epoch 18. We initialize the EfficientDet backbone using weights from COCO pretraining while the other layers are initialized using the Kaiming-He [10] initialization strategy.

4.3. Quantitative Results

Since we are the first to propose a method for self-supervised BEV segmentation, we benchmark SkyEye against EfficientPS [28] + IPM [25] as well as 5 fully-supervised approaches, namely Translating Images Into Maps (TIIM) [33], Variational Encoder Decoder (VED) [24], View Parsing Network (VPN) [30], Pyramid Occupancy Network (PON) [32], and PanopticBEV (PoBEV) [6]. We train the baseline models on our dataset using the open source code provided by the authors after minimally adapting them to handle the different input size, output size, and number of output semantic classes. We ensure fair comparison by adhering as closely as possible to the training protocols outlined in their respective publications. Tab. 1 presents the results of this evaluation using the class-wise Intersection-over-Union (IoU) and overall mean IoU (mIoU) metrics for the KITTI-360 dataset.

We observe that our proposed SkyEye model outperforms 5 of 6 baseline models by more than 3.65 pp and performs on par with the state-of-the-art fully-supervised approach PoBEV while not using any form of BEV ground truth supervision. We further note that our approach exceeds the baselines by up to 3.66 pp on all static classes such as *road*, *sidewalk*, *building*, and *terrain*. This superior performance for static classes can largely be attributed to our consistency-based implicit supervision which enables the network to infer spatially consistent features as well as reason about occluded and distant regions. At the same time, we observe that our model demonstrates inferior performance on dy-

Table 1. Evaluation of BEV semantic mapping on the KITTI-360 dataset. All metrics are reported in $[\%]$.

Method	BEV GT	Road	Sidewalk	Building	Terrain	Person	2-Wheeler	Car	Truck	mIoU
IPM [25]	✗	53.03	24.90	15.19	32.31	0.20	0.36	11.59	1.90	17.44
TIIM [33]	✓	63.08	28.66	13.70	25.94	0.56	**6.45**	33.31	8.52	22.53
VED [24]	✓	65.97	35.41	37.28	34.34	0.13	0.07	23.83	8.89	25.74
VPN [30]	✓	69.90	34.31	33.65	40.17	0.56	2.26	27.76	6.10	26.84
PON [32]	✓	67.98	31.13	29.81	34.28	2.28	2.16	37.99	8.10	26.72
PoBEV [6]	✓	70.14	35.23	34.68	40.72	2.85	5.63	**39.77**	**14.38**	30.42
SkyEye (Ours)	✗	**71.39**	**37.62**	**37.48**	**44.38**	**4.73**	4.72	32.73	10.84	**30.49**

namic classes such as *car* and *truck* as compared to PoBEV, reporting nearly $7\,\mathrm{pp}$ and $3.5\,\mathrm{pp}$ lower on *car* and *truck* respectively. This result is caused by the use of only a forward-facing FV image which impedes the network from reasoning about the shape and extent of various objects. Further, the sparsity of points on dynamic objects in distant regions hinders the generation of accurate pseudolabels and negatively impacts the performance of our pipeline. Nevertheless, our approach still extracts strong features for dynamic objects from both implicit and explicit supervision which allows it to outperform IPM, VED, and VPN, and be on par with TIIM.

4.4. Ablation Study

In this section, we study the impact of various components of our self-supervised pipeline using an ablation study on the KITTI-360 dataset. To this end, we perform two ablative experiments to independently analyze the impact of implicit and explicit supervision on the overall performance.

Implicit Supervision: In this experiment, we quantify the impact of implicit supervision on model performance by comparing the IoU metrics obtained when our model *with* and *without* implicit supervision is trained on different percentages of BEV pseudolabels. We thus define five percentage splits, i.e., 0.1%, 1%, 10%, 50%, and 100% of BEV pseudolabels, and accordingly sample a fixed subset of images given the split percentage. In these experiments, we use all FV images from the training split for model pretraining. We ensure equal representation of all scenes in the dataset by independently sampling the given percentage of images from each of the scenes. We also ensure model convergence for all percentage splits by increasing the number of epochs for splits having a lower percentage of BEV pseudolabels. Lastly, we also train the state-of-the-art fully-supervised approach, PoBEV, on the same percentage splits of BEV ground truth labels to act as a benchmark for evaluating the performance of our approach. Note that these approaches are evaluated using the same set of BEV ground truth labels. Tab. 2 presents the results for all percentage splits. We further present results of SkyEye trained using different splits of BEV ground truth labels in Sec. S.3 of the supplementary.

We observe that our model with implicit supervision (SkyEye) significantly outperforms both PoBEV as well as our model without implicit supervision (SkyEye*) by more than $7\,\mathrm{pp}$ for extremely low percentage splits of 0.1% and 1%. At such low sample counts, PoBEV suffers from the lack of BEV ground truth data while SkyEye is able to leverage FV training to generate good results in BEV. A large part of the gain can be attributed to the better segmentation of static classes which is a direct consequence of the robust supervision generated from the warping step of implicit training. We also observe that SkyEye* outperforms PoBEV by more than $1.5\,\mathrm{pp}$ at these percentages which highlights the training efficiency of our network architecture. Further, we observe that with only 10% of the labels, SkyEye almost matches the state-of-the-art result, while SkyEye* and PoBEV are $2.09\,\mathrm{pp}$ and $2.66\,\mathrm{pp}$ away respectively. This observation further emphasizes the benefit of incorporating implicit supervision into the training procedure to circumvent the need for expensive BEV ground truth annotations. However, from this percentage split onwards, we observe that PoBEV starts outperforming SkyEye for the *car* class which can be attributed to the better and consistent supervision of BEV ground truth labels. As the percentage of BEV samples increases, all approaches converge to a similar mIoU score since the labels in BEV compensate for the privileged information supplied by implicit supervision. However, we still note that SkyEye outperforms PoBEV for all static classes which demonstrates the ability of implicit supervision to positively augment the training procedure even in the presence of 100% of BEV labels. To further demonstrate the impact of implicit supervision on the trained model, we provide further results for every percentage split in Sec. S.3 and Sec. S.5 of the supplementary.

Explicit Supervision: In this experiment, we study the influence of various components of our BEV pseudolabel generation pipeline by removing each component from the overall solution proposed in Sec. 3.3 and analyzing the change in the overall model performance. Tab. 3 outlines the results of this ablation study. The first row of Tab. 3 shows the results of the model trained with all components of our pseudolabel generation pipeline and acts as a baseline for evaluating the impact of each constituent module. We observe in the second row that frame accumulation forms the most important step of the pseudolabel generation pipeline whose removal results in a drop of $7.23\,\mathrm{pp}$. Frame accumulation improves the density of static regions and helps the model reason about far-away and occluded regions. In the third row, we replace the depth-based lifting of static classes for

Table 2. Ablation study on the impact of Implicit Supervision on the overall network performance. All scores are reported on the KITTI-360 dataset.

BEV (%)	Method	BEV GT	Implicit	Epochs	Road	Sidewalk	Building	Terrain	Person	2-Wheeler	Car	Truck	mIoU
0.1	PoBEV	✓	-	300	54.73	19.08	22.63	5.40	0.00	0.00	3.81	0.00	13.21
	SkyEye	✗	✗		55.20	18.42	20.95	11.63	0.00	0.00	15.53	0.00	15.22
	SkyEye	✗	✓		**68.49**	**31.11**	**32.98**	**29.92**	0.00	0.00	**19.19**	0.00	**22.71**
1	PoBEV	✓	-	100	61.70	17.10	27.81	26.72	0.07	0.36	21.51	0.84	19.51
	SkyEye	✗	✗		64.25	22.43	32.20	24.41	0.44	0.00	24.09	0.69	21.06
	SkyEye	✗	✓		**72.00**	**33.76**	**37.59**	**38.75**	**3.77**	**1.81**	**28.04**	**9.53**	**28.15**
10	PoBEV	✓	-	50	70.00	32.75	**38.07**	34.43	0.80	3.33	**34.46**	9.25	27.89
	SkyEye	✗	✗		70.44	33.88	33.74	41.66	3.47	3.83	31.54	9.14	28.46
	SkyEye	✗	✓		**72.40**	**37.06**	36.89	**43.67**	**3.90**	**4.20**	31.05	**9.86**	**29.88**
50	PoBEV	✓	-	30	**72.09**	35.64	36.64	42.41	1.61	3.92	**41.41**	9.77	30.44
	SkyEye	✗	✗		71.93	33.59	36.43	42.63	4.05	4.30	31.44	**12.76**	29.64
	SkyEye	✗	✓		71.85	**37.43**	**38.76**	**44.15**	**5.07**	**4.54**	31.07	11.54	**30.55**
100	PoBEV	✓	-	20	70.14	35.23	34.69	40.71	2.85	**5.63**	**39.78**	**14.38**	30.43
	SkyEye	✗	✗		71.00	36.38	**37.76**	44.13	4.47	4.37	30.98	12.76	30.23
	SkyEye	✗	✓		**71.39**	**37.62**	37.48	**44.38**	**4.73**	4.72	32.73	10.84	**30.49**

Table 3. Ablation study on the efficacy of various constituent components of Explicit Supervision. All results are reported on the KITTI-360 dataset.

Accumulation	Depth	Clustering	BBox	Road	Sidewalk	Building	Terrain	Person	2-Wheeler	Car	Truck	mIoU
✓	✓	✓	✓	71.39	**37.62**	37.48	44.38	**4.73**	**4.72**	**32.73**	10.84	**30.49**
✗	✓	✓	✓	60.27	26.75	19.89	41.22	0.44	1.43	26.06	10.01	23.26
✓	✗	✓	✓	66.28	33.43	27.24	38.95	4.62	3.90	21.04	9.08	25.57
✓	✓	✗	✗	72.68	37.32	**37.60**	**44.91**	2.17	2.49	29.96	**11.17**	29.79
✓	✓	✓	✗	**72.73**	37.42	37.30	44.54	0.00	0.00	25.95	9.25	28.40

Table 4. Evaluation of model generalizability across datasets. Models pretrained on KITTI-360 are evaluated on Waymo.

Method	TIIM	VED	VPN	PON	PoBEV	SkyEye (Ours)
mIoU	16.53	14.02	13.52	12.05	16.94	**22.57**

pseudolabel creation with the IPM algorithm and observe a drop in performance of both static and dynamic classes. We attribute this behavior to the strong correlation between static and dynamic classes as they are predicted in a joint manner within a single map. The last two rows indicate that the outlier removal and incorporation of prior knowledge on dynamic classes yield a large performance gain w.r.t. cars (7.23 pp and 6.78 pp respectively) but have a minor impact on static classes and trucks.

4.5. Generalization Experiments

We evaluate the generalization ability of SkyEye by evaluating the best model from the KITTI-360 dataset on the Waymo dataset and comparing its performance to that of the other baselines. VED, VPN, PON, and PoBEV use image dimensions as channel count in their learnable layers which forces the Waymo image size to be equal to that of KITTI-360 (1408×384). TIIM and SkyEye are agnostic to the image size and we thus report the result obtained when using images of size 512×352 which aligns the Waymo camera intrinsics with that of KITTI-360. Tab. 4 presents the results of this experiment. We observe that our approach generalizes significantly better to the Waymo dataset as compared to other baselines outperforming them by 5.63 pp. This large gain in mIoU can be credited to the superior geometric reasoning and world modeling of our approach which is largely facilitated by our self-supervised learning framework. Our implicit supervision guides the network to learn geometrically coherent features which helps it in generalizing well across different datasets. On the contrary, the BEV ground truths do not enforce geometric consistency which results in the poor performance of the fully supervised baselines.

4.6. Qualitative Results

We further evaluate SkyEye by qualitatively comparing it with the state-of-the-art fully supervised approach, PoBEV, in Fig. 5. We observe from Fig. 5(a) that both PoBEV and SkyEye are able to capture the characteristics of close regions and are also able to localize nearby vehicles to a high accuracy. Further, as evident from the error/improvement maps in the last column, our approach precisely captures the contour of the sidewalk for near as well as distant regions. A similar observation can be made in Fig. 5(b) where our model accurately predicts the curve of the road over long distances, but PoBEV fails to do so and instead confuses road with sidewalk. This superiority in inferring static elements of the scene can be attributed to implicit supervision which encourages the network to learn consistent features over long horizons. Fig. 5(c, d) demonstrate that our network is able to successfully localize a large number of vehicles in the scene. These images also depict a limitation of SkyEye, i.e., the failure to predict cars in distant regions. This can be attributed to the extreme sparsity of dynamic objects in distant regions which results in the generation of sub-optimal BEV pseudolabels. This limitation, however, is not faced

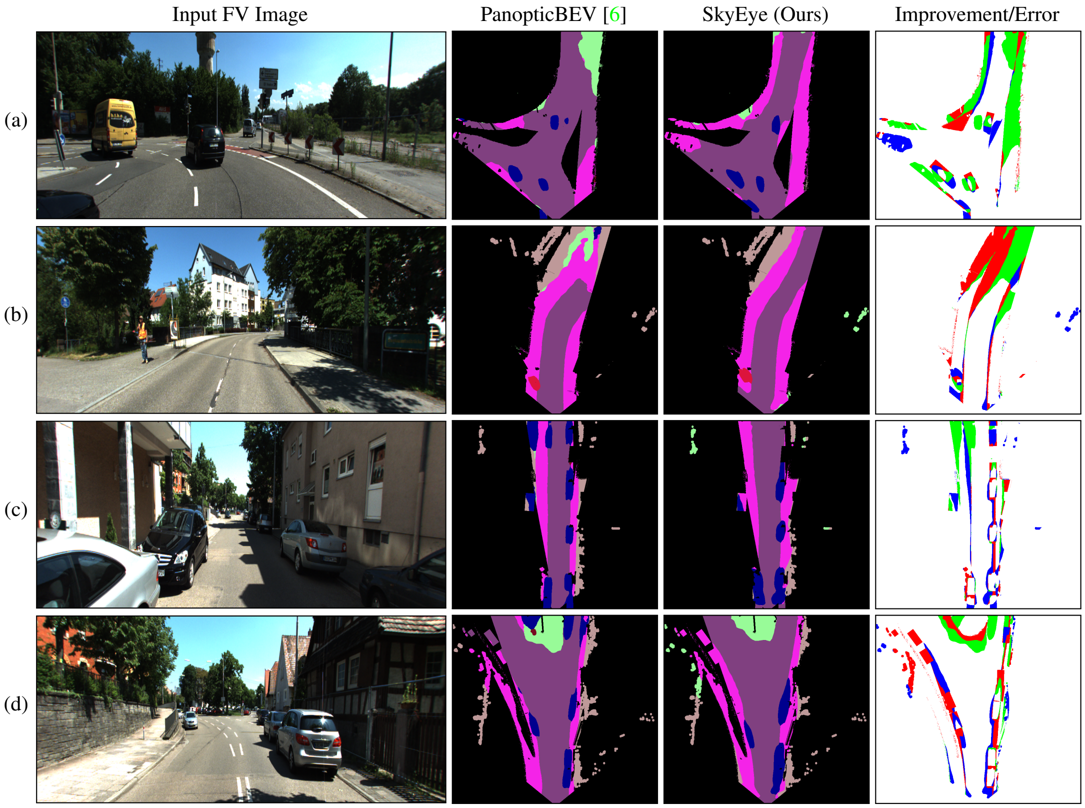

Figure 5. Qualitative results of our self-supervised framework SkyEye in comparison with PanopticBEV [6] on the KITTI-360 dataset. We also show the Improvement/Error map which shows pixels misclassified by PanopticBEV and correctly predicted by SkyEye in green, pixels misclassified by SkyEye and correctly by PanopticBEV in blue, and pixels misclassified by both models in red.

by PoBEV which uses BEV ground truth labels and thus receives reliable supervision throughout the BEV image. We also observe that our network accurately estimates the contours of *road*, *sidewalk*, and *terrain*, thus highlighting the benefit of implicit supervision. We provide further qualitative results of our approach in Sec. S.5 of the supplementary.

4.7. Discussion of Limitations

While our self-supervised approach, SkyEye, performs on par with fully supervised state-of-the-art approaches, it is subject to three kinds of limitations. Firstly, the reliance on temporal context can deteriorate its performance in highly dynamic scenes where moving objects may produce artifacts in both the generated pseudolabels and the implicit supervision signal. Here, explicit handling of moving objects can help minimize these effects. Secondly, our pseudolabels suffer from incorrect object extent and object heading estimates, especially when dealing with far-away objects. Lastly, perspective distortion limits the spatial observability of the scene for distant regions. This, however, is a longstanding limitation of camera-based methods that can be overcome using a longer temporal baseline in conjunction with a dedicated dynamic object handling strategy.

5. Conclusion

In this paper, we present SkyEye, the first self-supervised approach to generate a semantic map in BEV using a single FV monocular image, thus alleviating the need for expensive BEV semantic ground truth annotations. Our approach relies on only FV image sequences and their corresponding FV semantic annotations to generate two modes of supervision, namely, implicit supervision and explicit supervision. Using extensive evaluations on the KITTI-360 dataset, we demonstrate that SkyEye performs on par with the state-of-the-art fully supervised BEV approaches while already achieving competitive performance with only 1% of pseudolabels in the BEV. Future work includes relaxing the requirement for FV semantic ground truth labels and instead relying on coarse FV predictions from a generic pre-trained FV semantic network or leveraging scene knowledge from unsupervised FV semantic segmentation approaches.

Acknowledgements: This work was partly funded by the German Research Foundation (DFG) Emmy Noether Program grant number 468878300, the Bundesministerium fur Bildung und Forschung (BMBF) grant number FKZ 16ME0027, a CAPES-Alexander von Humboldt Foundation fellowship, and a hardware grant from NVIDIA.

References

[1] Borna Bešić and Abhinav Valada. Dynamic object removal and spatio-temporal rgb-d inpainting via geometry-aware adversarial learning. *IEEE Transactions on Intelligent Vehicles*, 7(2):170–185, 2022. 2

[2] Pramit Dutta, Ganesh Sistu, Senthil Yogamani, Edgar Galván, and John McDonald. Vit-bevseg: A hierarchical transformer network for monocular birds-eye-view segmentation. *arXiv preprint arXiv:2205.15667*, 2022. 2

[3] 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. 4

[4] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 2

[5] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3828–3838, 2019. 2, 3, 4

[6] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. *IEEE Robotics & Automation Letters*, 7(2):1968–1975, 2022. 1, 2, 5, 6, 8, 11, 12, 13

[7] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pages 6391–6400, 2019. 2

[8] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 2

[9] Liming Han, Yimin Lin, Guoguang Du, and Shiguo Lian. Deepvio: Self-supervised deep learning of monocular visual inertial odometry using 3d geometric constraints. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6906–6913, 2019. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 5

[11] Derek Hoiem, Alexei A Efros, and Martial Hebert. Geometric context from a single image. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 654–661. IEEE, 2005. 2

[12] Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Learning kinematic feasibility for mobile manipulation through deep reinforcement learning. *IEEE Robotics and Automation Letters*, 6(4):6289–6296, 2021. 1

[13] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7396–7405, 2020. 2

[14] Juana Valeria Hurtado, Laura Londoño, and Abhinav Valada. From learning to relearning: A framework for diminishing bias in social robot navigation. *Frontiers in Robotics and AI*, 8:650325, 2021. 1

[15] Juana Valeria Hurtado and Abhinav Valada. Semantic scene segmentation for robotics. In *Deep Learning for Robot Perception and Cognition*, pages 279–311. Elsevier, 2022. 1

[16] Isinsu Katircioglu, Bugra Tekin, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Learning latent representations of 3d human pose with deep neural networks. *International Journal of Computer Vision*, 126(12):1326–1341, 2018. 2

[17] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 876–885, 2017. 1

[18] Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9730–9740, 2021. 2

[19] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An online HD map construction and evaluation framework. In *Int. Conf. on Robotics & Automation*, pages 4628–4634, 2022. 1, 2

[20] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 5

[21] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6648–6657, 2020. 2

[22] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 2

[23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. 1, 2

[24] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. *IEEE Robotics and Automation Letters*, 4(2):445–452, 2019. 2, 5, 6, 11

[25] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991. 5, 6, 11

[26] Claudio Mello Jr, Bryan Moreira, Paulo Evald, Paulo Drews-Jr, and Silvia Botelho. Underwater enhancement based on a self-learning strategy and attention mechanism for high-intensity regions. *Computers & Graphics*, 107:264–276, 2022. 2

[27] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object

detection from lidar point cloud. In *European Conference on Computer Vision*, pages 515–531. Springer, 2020. 1

[28] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021. 3, 5

[29] Mong Him Ng, Kaahan Radia, Jianfei Chen, Dequan Wang, Ionel Gog, and Joey Gonzalez. BEV-Seg: Bird's eye view semantic segmentation using geometry and semantic point cloud. *CVPR Workshop on Scalability in Autonomous Driving*, 2020. 1

[30] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. 2, 5, 6, 11

[31] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conf. on Computer Vision*, 2020. 2

[32] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2020. 2, 5, 6, 11

[33] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9200–9206. IEEE, 2022. 2, 5, 6, 11

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2

[35] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. 4

[36] Samuel Schulter, Menghua Zhai, Nathan Jacobs, and Manmohan Chandraker. Learning to look around objects for top-view representations of outdoor scenes. In *European Conf. on Computer Vision*, pages 787–802, 2018. 1

[37] Inkyu Shin, Dong-Jin Kim, Jae Won Cho, Sanghyun Woo, Kwanyong Park, and In So Kweon. Labor: Labeling only if required for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8588–8598, 2021. 1

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 5

[39] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 3

[40] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Continual slam: Beyond lifelong simultaneous localization and mapping through continual learning. *arXiv preprint arXiv:2203.01578*, 2022. 2

[41] Jan Ole von Hartz, Eugenio Chisari, Tim Welschehold, and Abhinav Valada. Self-supervised learning of multi-object keypoints for robotic manipulation. *arXiv preprint arXiv:2205.08316*, 2022. 2

[42] Jiayu Yang, Jose M Alvarez, and Miaomiao Liu. Self-supervised learning of depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7526–7534, 2021. 2

[43] Abdelrahman Younes, Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Catch me if you hear me: Audio-visual navigation in complex unmapped environments with moving sounds. *IEEE Robotics and Automation Letters*, 2023. 1

[44] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022. 2