
Nikhil Gosala1*  Kürsat Petek1*  Paulo L. J. Drews-Jr1,2  Wolfram Burgard3  Abhinav Valada1
1University of Freiburg  2Federal University of Rio Grande  3University of Technology Nuremberg

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 the 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.

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. 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, utilizes 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 as 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 has found use in applications such as novel view synthesis and 3D scene 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 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 ($L_{fv}$, 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 ($L_{bev}$, Sec. 3.3). The final loss is thus computed as:

$$\mathcal{L} = L_{fv} + L_{bev}$$ (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 $1 \times 1$ convolution. Similarly, we generate the BEV logits by flattening the 3D voxel grid orthographically along the height dimension and passing it through a $1 \times 1$ 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 $3 \times 3$ 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
We compute the cross entropy loss between the FV semantic maps at multiple time steps. To this end, we predict the FV semantic maps for the initial \( t_0 \) as well as future time steps \( (t_1, \ldots, t_n) \) 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 \( I_0 \) to generate the intermediate 3D voxel grid representation \( V_0 \). This voxel grid is perspectively distorted using the camera intrinsics and processed along the depth dimension to generate the FV semantic prediction \( S_0 \). Perspective distortion of the voxel grid prior to processing it in FV is crucial to prevent implicit supervision from distorting the voxel grid. Parallely, we transform \( V_0 \) to generate the voxel grids \( V_{0\rightarrow i} \) for future time steps \( (t_1, \ldots t_n) \) using the relative ego poses \( T_{0\rightarrow i} \) between the initial and future time steps. We then use the generated pseudo voxel grids to infer the future FV semantic predictions \( S_{0\rightarrow i}, S_{0\rightarrow 2}, \ldots, S_{0\rightarrow n} \). 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 \( L_{fv} \) by accumulating the losses for each time step \( L_{fv,i} \) as:

\[
L_{fv} = \sum_{i=0}^{n} L_{fv,i} = \sum_{i=0}^{n} w_i CE(S_{0\rightarrow i}, S_i),
\]

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 that consists of 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 \([35]\) to incorporate prior knowledge on dynamic objects, 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 \([3]\) 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 \( W \) of FV images and their corresponding semantic ground truths. Fig. 4 shows an example of each proposed step. We first predict the depth map \( D_i \) for each FV image \( I_i \in W \) 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 \( P_0 = \bigcup_{i \in W} P_{k_i=0} \). 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 bounding boxes around dynamic objects (Fig. 2).

Firstly, we orthographically project points belonging to static classes to generate a sparse BEV map. We then densify the sparse BEV map by applying a series of morphological dilate and erode operations to generate the first set of labels \( \hat{B}^s \) for static classes. Secondly, we try to mitigate the lack of observability of the shape of dynamic objects by incorporating prior knowledge and fitting bounding 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 the presence of 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 \( M \) of \( C \) clusters, where \( M = \{m_j, j = 1\ldots C\} \). The points belonging to the \( M \) clusters are then orthographically projected into the BEV space and an ellipse \( E = \{x_c, y_c, a, b, \theta\} \), which serves as an analytical and differentiable replacement of a bounding box, is fit 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 \( \theta \) define the position, extents, and orientation of the bounding box in BEV, respectively. This procedure generates a second BEV map \( \hat{B}^d \) which is then overlaid on \( \hat{B}^s \) to generate the final pseudolabel map \( \hat{B}^{pl} \) 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:

\[
L_{bev} = CE(\hat{B}^{pl}, \hat{B}).
\]
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 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 pre-training 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 pre-trained model on all samples in the Waymo validation split.

4.2. Training Protocol

We train SkyEye on images of size $1408 \times 384$ 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 dynamic classes such as car and truck as compared to PoBEV, reporting nearly 7 pp and 3.5 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 TIM.

### 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. 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. Tab. 2 presents the results for all the percentage splits.

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 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 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 pp and 2.66 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 qualitative results for every percentage split in the supplementary material.

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

<table>
<thead>
<tr>
<th>BEV (%)</th>
<th>Method</th>
<th>BEV GT</th>
<th>Implicit</th>
<th>Epochs</th>
<th>Road</th>
<th>Sidewalk</th>
<th>Building</th>
<th>Terrain</th>
<th>Person</th>
<th>2-Wheeler</th>
<th>Car</th>
<th>Truck</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>PoBEV</td>
<td>✓</td>
<td>-</td>
<td>300</td>
<td>54.73</td>
<td>19.08</td>
<td>22.63</td>
<td>5.40</td>
<td>0.00</td>
<td>0.00</td>
<td>3.81</td>
<td>0.00</td>
<td>13.21</td>
</tr>
<tr>
<td></td>
<td>SkyEye</td>
<td>✓</td>
<td>-</td>
<td></td>
<td>55.20</td>
<td>18.42</td>
<td>20.95</td>
<td>11.63</td>
<td>0.00</td>
<td>0.00</td>
<td>15.53</td>
<td>0.00</td>
<td>15.22</td>
</tr>
<tr>
<td></td>
<td>SkyEye</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>68.49</td>
<td>31.11</td>
<td>32.98</td>
<td>29.92</td>
<td>0.00</td>
<td>0.00</td>
<td>19.19</td>
<td>0.00</td>
<td>22.71</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Accumulation</th>
<th>Depth</th>
<th>Clustering</th>
<th>BBox</th>
<th>Road</th>
<th>Sidewalk</th>
<th>Building</th>
<th>Terrain</th>
<th>Person</th>
<th>2-Wheeler</th>
<th>Car</th>
<th>Truck</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>71.39</td>
<td>37.62</td>
<td>37.48</td>
<td>44.38</td>
<td>4.73</td>
<td>4.72</td>
<td>32.73</td>
<td>10.84</td>
<td>30.49</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>60.27</td>
<td>26.75</td>
<td>19.89</td>
<td>41.22</td>
<td>0.44</td>
<td>0.40</td>
<td>26.06</td>
<td>10.01</td>
<td>23.26</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>66.28</td>
<td>33.43</td>
<td>27.24</td>
<td>38.95</td>
<td>4.62</td>
<td>3.90</td>
<td>21.04</td>
<td>9.08</td>
<td>25.57</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>72.68</td>
<td>37.32</td>
<td>37.60</td>
<td>44.91</td>
<td>2.17</td>
<td>2.49</td>
<td>29.96</td>
<td>11.17</td>
<td>29.79</td>
</tr>
</tbody>
</table>

| ✓            | ✓     | ✓          | ✓    | 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 pre-trained on KITTI-360 are evaluated on Waymo.

<table>
<thead>
<tr>
<th>Method</th>
<th>TiM</th>
<th>VED</th>
<th>VPN</th>
<th>PON</th>
<th>PoBEV</th>
<th>SkyEye (Ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>16.53</td>
<td>14.02</td>
<td>13.52</td>
<td>12.05</td>
<td>16.94</td>
<td>22.57</td>
</tr>
</tbody>
</table>

removal results in a drop of 7.23 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 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). TiM 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 our network, i.e., the failure to predict cars in distant regions. This behavior can be attributed to the extreme sparsity of
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 two 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, perspective distortion limits the spatial observability of the scene for distant regions. This, however, is a long-standing 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 funded by the German Research Foundation (DFG) Emmy Noether Program grant number 468878300 and a hardware grant from NVIDIA.
References


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


- Supplementary Material -

Nikhil Gosala∗, Kürsat Petek∗, Paulo L. J. Drews-Jr, Wolfram Burgard, and Abhinav Valada

In this supplementary material, we present additional experiments to study the performance of SkyEye. Specifically, we analyze the computational efficiency of our model in Sec. S.1 and perform an additional ablative experiment to study the characteristics of our contribution in Sec. S.2. We also present further qualitative results to demonstrate the performance of SkyEye on the KITTI-360 dataset in Sec. S.3.

S.1. Evaluation of Model Efficiency

In this section, we compare the efficiency of SkyEye with that of the other baselines on the KITTI-360 dataset. Tab. S.1 presents the results of this comparison on an nVidia GTX 3090 GPU. We observe that SkyEye is the most efficient of all baselines in terms of the number of learnable parameters, using 7.73 million fewer parameters as compared to the state-of-the-art fully supervised approach, PoBEV [6]. We also note that SkyEye uses significantly fewer Multiply-Accumulate (MAC) operations as compared to PoBEV. A large part of this efficiency can be attributed to the use of a 3D voxel grid representation to jointly represent features of both FV and BEV, thus alleviating the need for separate representations in FV and BEV. Further, we observe that our network design translates into a nearly 3-fold reduction in runtime as compared to PoBEV, requiring only 77.84 ms per sample and allowing SkyEye to be used in many real-time applications.

Table S.1. Comparison of model efficiency on the KITTI-360 dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th># Params (M)</th>
<th>MAC (G)</th>
<th>Runtime (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPM [25]</td>
<td>27.83</td>
<td>32.51</td>
<td>71.65</td>
</tr>
<tr>
<td>VED [24]</td>
<td>169.20</td>
<td>391.49</td>
<td>25.42</td>
</tr>
<tr>
<td>VPN [30]</td>
<td>23.01</td>
<td>55.45</td>
<td>8.68</td>
</tr>
<tr>
<td>PON [32]</td>
<td>91.06</td>
<td>657.63</td>
<td>164.08</td>
</tr>
<tr>
<td>TIIM [33]</td>
<td>40.72</td>
<td>1290.16</td>
<td>193.82</td>
</tr>
<tr>
<td>PoBEV [6]</td>
<td>22.33</td>
<td>272.68</td>
<td>77.84</td>
</tr>
<tr>
<td>Ours</td>
<td>14.60</td>
<td>219.94</td>
<td>77.84</td>
</tr>
</tbody>
</table>

S.2. Additional Ablation Experiments

In this section, we perform an ablation experiment to analyze the impact of different FV window sizes (WS) used during implicit supervision on the overall performance of the model. To this end, we perform implicit supervision using windows of size 0, 6, 10, 14, and 20, and subsequently refine these pre-trained models using 1% of BEV pseudo labels. We choose 1% of BEV pseudo labels to better highlight the impact of window sizes on the overall performance of the model. Tab. S.2 presents the results of this ablation study. We observe that a window size of 10 generates the best mIoU score across all the tested window sizes, marginally outperforming a window size of 6 by 0.12 pp. We also observe that larger window sizes report worse performance across almost all classes. Thus, we use WS = 10 in our SkyEye framework and perform all experiments with this window size.

S.3. Additional Qualitative Results

In this section, we present additional qualitative results on the KITTI-360 dataset and also present the BEV semantic maps obtained when our SkyEye model is trained using different percentages of BEV pseudo labels.

S.3.1. BEV Semantic Mapping

We qualitatively demonstrate the performance of our model by comparing the output from our network to those obtained from the state-of-the-art fully supervised approach, PoBEV [6]. We also show an Error/Improvement map in the rightmost column to highlight the difference in predictions between both approaches. Fig. S.1 presents the semantic BEV map predictions obtained from both networks. We observe from the figures that our approach is mostly on par with PoBEV across a wide range of scenarios from straight roads with multiple parked cars, to curved roads, and to complex intersections. As already noted in the main paper, we observe from Fig. S.1(d, e, f) that SkyEye performs significantly better on static regions as compared to PoBEV which can be confirmed by looking at the large swathes of green in the last column. Our model is able to estimate the extent of roads and sidewalks better than PoBEV, and this improvement in performance can be attributed to training using implicit supervision which encourages the model to learn spatially coherent representations. We also observe from Fig. S.1(a, b, c) that our model is able to accurately estimate the locations of multiple cars in the scene and these images qualitatively look very similar to that of PoBEV.
Table S.2. Ablation study on the impact of Window Size on the overall performance of the model. This experiment uses only 1% of BEV pseudo labels to highlight the impact of window size. All scores are reported on the KITTI-360 dataset.

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Road</th>
<th>Sidewalk</th>
<th>Building</th>
<th>Terrain</th>
<th>Person</th>
<th>2-Wheeler</th>
<th>Car</th>
<th>Truck</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>71.82</td>
<td>33.27</td>
<td>36.00</td>
<td>41.23</td>
<td>2.80</td>
<td>0.00</td>
<td>24.97</td>
<td>5.42</td>
<td>26.94</td>
</tr>
<tr>
<td>6</td>
<td>71.97</td>
<td>35.63</td>
<td>36.86</td>
<td>39.07</td>
<td>2.93</td>
<td>0.00</td>
<td>28.23</td>
<td>9.55</td>
<td>28.03</td>
</tr>
<tr>
<td>10</td>
<td>72.00</td>
<td>33.76</td>
<td>37.59</td>
<td>38.75</td>
<td>3.77</td>
<td>1.81</td>
<td>28.04</td>
<td>9.53</td>
<td>28.15</td>
</tr>
<tr>
<td>14</td>
<td>71.81</td>
<td>32.87</td>
<td>36.93</td>
<td>38.47</td>
<td>3.02</td>
<td>2.87</td>
<td>28.58</td>
<td>7.93</td>
<td>27.81</td>
</tr>
<tr>
<td>20</td>
<td>71.21</td>
<td>35.16</td>
<td>35.77</td>
<td>38.26</td>
<td>3.59</td>
<td>2.35</td>
<td>28.91</td>
<td>5.16</td>
<td>27.43</td>
</tr>
</tbody>
</table>

Figure S.1. Additional Qualitative Results on the KITTI-360 dataset. The rightmost column shows the Improvement/Error map which depicts pixels misclassified by PoBEV and correctly predicted by SkyEye in green, pixels misclassified by SkyEye and correctly by PoBEV in blue, and pixels misclassified by both models in red.
Figure S.2. Qualitative results obtained when SkyEye is trained using 0.1%, 1%, 10%, 50% and 100% of BEV pseudolabels.
However, we note that when cars are extremely close to each other, our model sometimes stretches the extent of cars and merges multiple cars into one big blob. This is a limitation of our model and is largely a consequence of using only a forward-facing camera during implicit supervision which inhibits the model from extracting the extents of all four sides of vehicles. Furthermore, our method does not have access to the ground truth BEV map obtained using LiDAR data that is able to perceive both close and distant vehicles with higher precision. Nevertheless, our model performs on par with PoBEV without using any ground truth supervision in BEV, thus highlighting the impact of our self-supervised BEV semantic mapping framework.

**S.3.2. Different Percentages of BEV Pseudolabels**

In this section, we qualitatively evaluate the impact of using different percentages of BEV pseudolabels on the overall performance of our model. Fig. S.2 presents the results of this evaluation. We observe a very interesting trend across samples when training our model with different percentages of BEV pseudolabels. Across all images in Fig. S.2, we observe that our model gradually improves its reasoning about dynamic cars in the scene with an increase in the percentage of BEV pseudolabels. When using only 0.1% of pseudolabels, our model fails to identify any dynamic cars in the scene. Upon increasing the number of pseudolabels to 1%, we see that our model starts reasoning about the locations of dynamic objects but the cars look stretched and artificial. However, upon further increasing the percentage of BEV pseudolabels to 10%, our model predicts the locations as well as the extent of cars accurately. Further increase in the percentage of pseudolabels refines the predictions and better constrains the extent of vehicles, but no significant improvement can be observed between the BEV maps obtained using 10%, 50% and 100% of pseudolabels. This supports our findings in Tab. 2 of the main paper where we observe no large changes in mIoU scores when using 10%, 50%, and 100% of BEV pseudolabels to train the model.

Further, a special observation can be made when the scene is fully static. Fig. S.2(c, d, f) depict static scenes with multiple parked cars wherein we observe that our model with 0.1% of pseudolabels is already able to predict the locations of static cars with high accuracy. This accurate estimation of the location of static cars with very few pseudolabels can be attributed to the structure infused into the 3D voxel grid representation by implicit supervision. This strong structural signal thus enables the model to reason about the world in BEV even when the model is exposed to extremely sparse samples in BEV.