Ithaca365: Dataset and Driving Perception under Repeated and Challenging Weather Conditions

Carlos A. Diaz-Ruiz[1] Youya Xia[1] Yurong You[1] Jose Nino[1] Junan Chen[1]
Josephine Monica[1] Xiangyu Chen[1] Katie Luo[1] Yan Wang[1] Marc Emond[1]
Wei-Lun Chao[2] Bharath Hariharan[1] Kilian Q. Weinberger[1] Mark Campbell[1]
[1]Cornell University [2]The Ohio State University

Abstract

*Advances in perception for self-driving cars have accelerated in recent years due to the availability of large-scale datasets, typically collected at specific locations and under nice weather conditions. Yet, to achieve the high safety requirement, these perceptual systems must operate robustly under a wide variety of weather conditions including snow and rain. In this paper, we present a new dataset to enable robust autonomous driving via a novel data collection process — data is **repeatedly** recorded along a 15 km route under **diverse scene** (urban, highway, rural, campus), **weather** (snow, rain, sun), time (day/night), and **traffic conditions** (pedestrians, cyclists and cars). The dataset includes images and point clouds from cameras and LiDAR sensors, along with high-precision GPS/INS to establish correspondence across routes. The dataset includes road and object annotations using **amodal masks** to capture partial occlusions and 3D bounding boxes. We demonstrate the uniqueness of this dataset by analyzing the performance of baselines in amodal segmentation of road and objects, depth estimation, and 3D object detection. The repeated routes opens new research directions in object discovery, continual learning, and anomaly detection. Link to Ithaca365:* `https://ithaca365.mae.cornell.edu/`

1. Introduction

The self-driving car research community has made major advancements in computer vision and perception by relying on vast amounts of real world sensory datasets. To date, many datasets have been published [4, 6, 14, 18, 32], including some with benchmark challenges associated with different perception tasks. These tasks include 2D image detection [22, 30], depth estimation (stereo [5, 35] and monocular [11]), 3D object detection (stereo [19, 29, 34] and LiDAR [8, 36, 38]), semantic [39] and instance segmentation [20, 21]. The KITTI dataset [14] has been used for many of these benchmarks due to its comprehensive annotations and ground truths. However, like many of the others [9], the KITTI dataset was collected in only sunny/clear conditions.

There is a need in the research community for large-scale datasets with multi-modal sensor data (LiDAR, cameras, and GPS/IMU) in *adverse weather conditions* to be able to train and test the performance and robustness of object detectors, image segmentation, and depth estimation algorithms across varying conditions. The lack of data in these challenging conditions ultimately limits the generalizability of the perception approaches, and as such, limits the applicability of self-driving cars only to environments with nice weather conditions. Recently, a few datasets with more diverse traffic, scenes, and weather conditions have been published [4, 6, 24]. These datasets include rain, low light conditions at night, or very bright conditions, but do not include snowy conditions. One such dataset with snowy conditions is the Canadian Adverse Driving Conditions Dataset [26], a dataset collected in Waterloo. However, this dataset is primarily snowy conditions, and includes just 3D object labels.

Moreover, to our knowledge an amodal segmentation dataset with varying weather conditions does not exist. Amodal perception [20, 40] aims to perceive and understand the physical structure and semantics of occluded objects and scenes. Amodal perception is critical for autonomous driving as the ability to infer the whole shape of objects (e.g., other vehicles, pedestrians, and road) around a mobile platform allows for safer and more efficient navigation. This is especially useful in highly cluttered environments or complex traffic scenes, where determining safe, collision free paths based only on visible cues is difficult. The KINS dataset [28] augments KITTI [13] with amodal foreground masks and while [23] augments Cityscapes [9] with amodal background classes, but both of these are only in sunny and clear conditions. There are a few recent developments that address amodal segmentation [20, 23, 28]. Unfortunately, existing literature lacks progress in these areas, primarily due to the lack of large-scale, diverse datasets annotated with amodal masks.

In this paper we provide three major contributions. First, we release a large-scale, *weather and environmental di-*

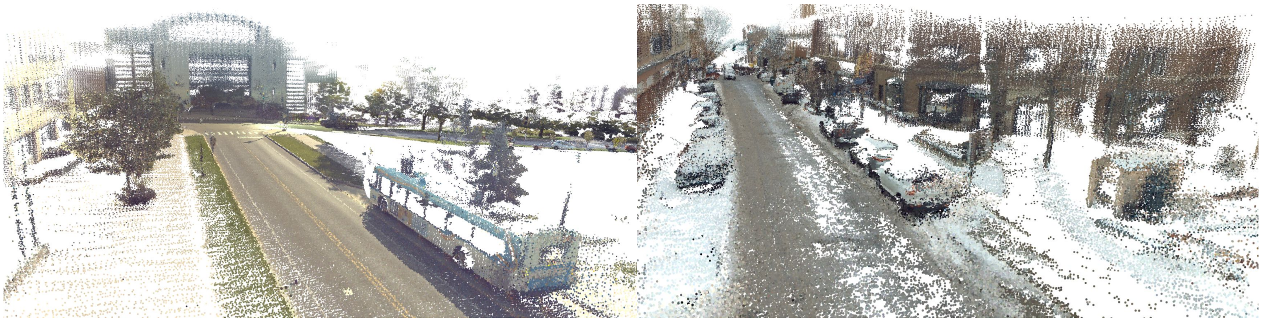


Figure 1. Point clouds generated using calibrated images, LiDAR points, and GPS/IMU data. Left: Point cloud from a sunny day while driving near buildings on campus. Right: Point cloud from a snow day (i.e., white on the road) through an urban portion of the route.

verse, *amodal* dataset obtained through *repeatedly* driving a 15km route over a 1.5 year period. The key attributes are:

- **weather diversity**: snow, rain, sunny, cloudy, night
- **environmental diversity**: urban traffic, highway, rural, pedestrian heavy university campus
- **amodal**: roads, cars,pedestrians occluded by snow,objects
- **repeated route**: 40 data collections on a 15km route

More than 680k frames were collected with LiDAR, images and GPS data. Figure 1 shows some 3D visualizations. Because of the repeatability of the route, amodal labeling of background classes such as road and even parked cars are easily facilitated across the extremely varied conditions without tedious annotation. Second, we suggest new metrics for the proper evaluation of amodal segmentation of 3D objects and background classes such as road. Finally, we develop baseline model architectures to highlight the utility of the proposed dataset for key self-driving car perception tasks: amodal background segmentation (road), amodal instance segmentation (car, pedestrians, cyclists, motorcyclists, truck, bus, etc.), 3D object detection, depth estimation across varying weather conditions, route types, and leveraging repeatability for unsupervised labeling.

2. Related Work

Autonomous driving datasets. The **KITTI** dataset [14] is one of the most popular auto driving datasets and has paved the way for the development of deep learning algorithms for 2D and 3D object detection, stereo depth estimation, and segmentation tasks; all data was collected in daylight and nice weather. The **Oxford RoboCar** dataset [24] contains stereo cameras, 2D and 3D LiDAR, GPS/IMU data; this dataset capture various weather conditions along the same route but only reports one route with snow conditions. Additionally, the dataset was released with a focus on localisation and thus does not include 2D/3D object or segmentation annotations. The recent **GROUNDED** dataset [25] is for localization in varying weather conditions, with a focus on ground penetrating radar. The **Canadian Adverse Driving Conditions (CADC)** dataset [26] includes 7K frames with 3D object labels in snow conditions with surround vision, LiDAR data, and ground truth motion. The **ApolloScape** open dataset [33] contains LiDAR, cameras, and GPS in cloudy and rainy conditions, along with brightly sunlit situations. The dataset includes large scale 3D car annotations and segmentation labels. The authors indicate that data in snow conditions will be added, but none has been released. There have been various other large-scale datasets: The **nuScenes** dataset [4], **Argoverse** dataset [6], **Waymo** open dataset [32], and **Lyft Level 5** dataset [18]. NuScenes has 40k annotated frames, with LiDAR, cameras and radar. Waymo is the largest dataset with 200k frames and 5 LiDARs, and cameras. Lyft, to our knolwedge, is the only dataset with 3D box labels along a repeated route, but mostly in nice weather (Palo Alto). Although large-scale, none of these datasets were collected in snowy conditions. Finally, while the Oxford and CADC datasets include data in snowy conditions, they do not include important amodal segmentations, i.e., no labels of invisible and occluded parts of objects. These datasets cannot be used for direct supervised amodal training or evaluation. Both KITTI and **Cityscapes** [9] have been augmented to have amodal segmentation labels for some classes. **KINS** [28] added labels of traffic participant classes to 14,991 images, and [23] annotated 500 test images for background classes in Cityscapes and 200 images in KITTI. KINS does not provide the road amodal ground truths addressed in this work, and both KITTI and CityScapes do not have very adverse weather or lighting conditions. There is a critical need in the community for a holistic dataset with amodal labels across adverse weather conditions to be able to train and test the performance and robustness of object detectors, image segmentors, and depth estimators.

Image and amodal segmentation. Image segmentation can be divided into instance segmentation and semantic segmentation. The former aims to identify each individual object instance. Mask R-CNN [16] is one representative algorithm. The latter aims to label each pixel with a semantic label. State-of-the-art models exploit multi-scale feature fusion (i.e., [39]) or self-attention (e.g., [37]).

Although both segmentation tasks have been studied extensively, they only reason about the visible parts of the observed scene and fail to learn the occluded semantic information. Thus, in recent years, amodal segmentation has gained research interest [20,28]. However, due to the lack of amodal labeled datasets, amodal segmentation has mostly focused on learning the invisible information using unsupervised algorithms [20], which suffer from grave outliers. KINS [28] is a pioneer amodal dataset for driving scenes; the paper also proposes an amodal baseline model with an occlusion branch. More recently, [23] proposes a Semantic Foreground Inpainting (SFI) model from weak supervision using a two-branch (background and foreground) architecture. The model aims to learn the amodal semantic background mask by inpainting the predicted foreground areas with nearby background features. We combine SFI with position and channel attention modules by [37], to develop a strong baseline for amodal road segmentation.

3. Creating the Dataset

3.1. The sensor platform

The data was collected using a self-driving car research platform, illustrated in Figure 2. A 2015 Lexus RX 450h is equipped with the following sensors and hardware:

- 4×Sekonix SF3325-100 cameras, 1928×1208, 30Hz, GMSL, RCCB Filter, 1/2.7”, 60° horizontal FOV
- 2×Velodyne Puck (VLP-16) LiDAR, 16 scan, 100m range, 30° vertical FOV, 10Hz
- 1×Ouster OS2-128 long range LiDAR, 128 scan, 240m range, 22.5° vertical FOV, 10Hz
- 1×Novatel PwrPak7D, dual GNSS-502 antenna, Epson G320N IMU, custom firmware for PTP timing, high precision fixes from PointOne
- 1×Nvidia AGX

The sensors and their locations on the vehicle are shown in Figure 2. We will release json files that specify extrinsic parameters to each sensor. NVIDIA’s recording tool (part of the AGX) was utilized to log the LiDARs at 10 Hz and cameras at 30 Hz. The Novatel GPS/INS data was logged using a PC running ROS, and time synchronized with the AGX through PTP and Novatel custom firmware. Time synchronization between cameras has been verified to 10s of microseconds. More details on synchronization between sensors have been included in Appendix A.2. LiDAR scans can be combined with GPS/INS to form accurate 3D reconstructions of the environment as shown in Figure 1.

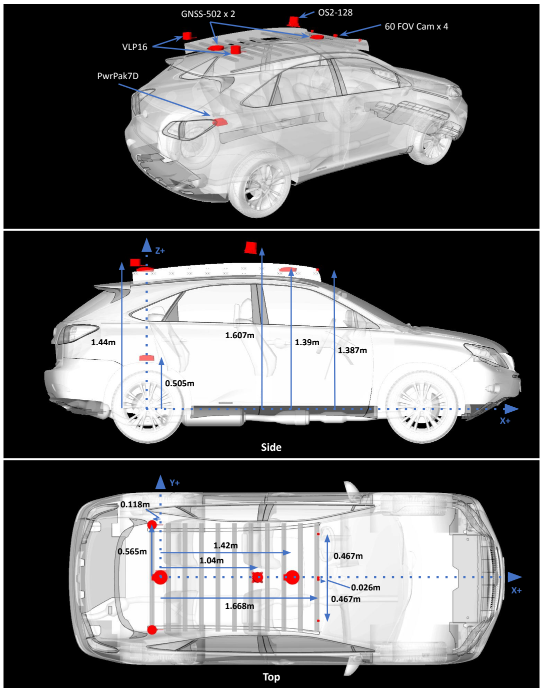


Figure 2. Sensors and mounting locations on the car.

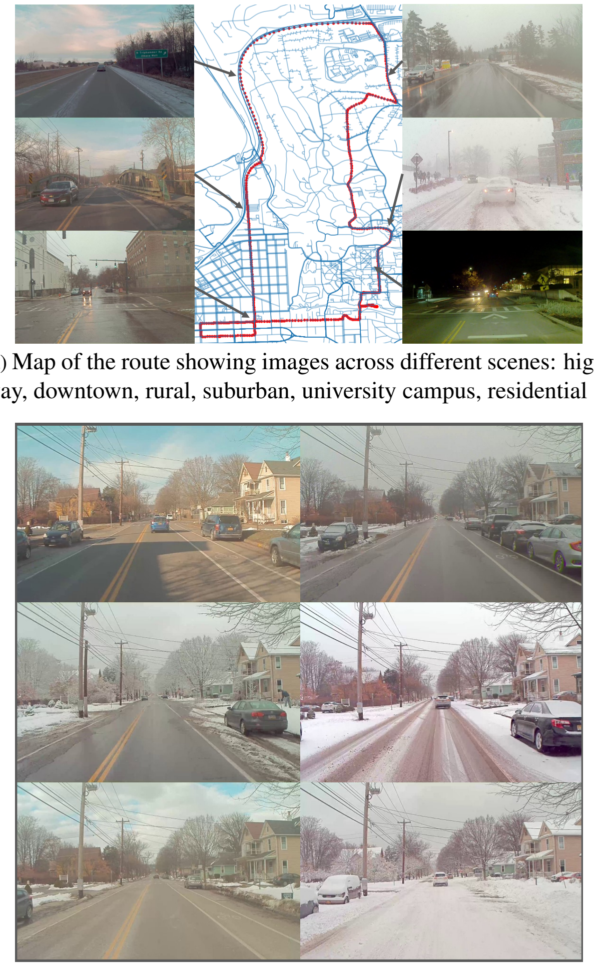
3.2. Route and data collection

A 15 km loop consisting of varying road types and surrounding environments was selected. The route includes university campus, downtown, highway, urban, residential, and rural areas. Figure 3a shows a map of the route with images captured at several locations. Driving was scheduled to capture data at different times of day, including night. Heavy snow situations were captured before and after roads were plowed. A key uniqueness of our dataset is that the same locations can be observed across different conditions; an example is shown in Figure 3b. A breakdown of the traversals under different conditions is shown in Figure 4.

3.3. Sensor calibration

Camera intrinsic, LiDAR to camera extrinsic. For each camera we provide camera intrinsic parameters. The camera intrinsic and distortion parameters were inferred by openCV [3] camera calibration functions and target checkerboards. The LiDAR to camera extrinsic matrix calibration was performed using multiple checkerboards at different locations/ranges in front of all sensors. Given the camera intrinsic and undistortion parameters, calibration matched the detected target points on checkerboards in LiDAR coordinates with the corresponding target points in camera coordinates. The calibration code along with calibration images will be released for custom analysis.

Lidar to IMU. We provide calibration between LiDAR and GPS/IMU sensors. LiDAR and GPS/IMU data was collected with large turning angles, and relative transformation (rotation, translation) in the LiDAR frame was estimated by performing Iterative Closest Point (ICP) between LiDAR point clouds. With the estimated transformation and pose readings from GPS/IMU (in IMU frame), we use a hand-



(a) Map of the route showing images across different scenes: highway, downtown, rural, suburban, university campus, residential

(b) Same location across varying weather conditions

Figure 3. Route and image visualizations

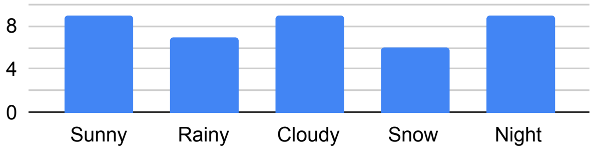


Figure 4. Histogram of conditions present in dataset.

eye calibration method [17] to obtain the relative transformation between LiDAR and IMU coordinates. Correction for the ego motion within LiDAR sensors has been implemented by interpolating the pose based on the time offset of laser firing during the LiDAR rotation, as specified by Ouster and Velodyne. This timing can be combined with GPS/INS data to compensate for the ego motion.

3.4. Amodal Labeling: Road

We developed a custom labeling tool to obtain the amodal masks for the road and objects. For road labels across different environmental conditions, such as snow covered roads, *we leveraged the repeated traversals of the same route.* Specifically, *road labels from nice weather are transferred to poor weather conditions, through point cloud road maps built using the GPS pose and LiDAR data.* The route/data was divided into 76 intervals. The point clouds were then projected into birds-eye-view (BEV) and the road was labeled using a polygon annotator. Once the road has been labeled in the BEV, which yields the 2D road boundary, the height of the road was determined by decomposing the polygon into smaller 150 m^2 polygons and fitting a plane to the points within the polygon boundary that are within a 1.5 m threshold of the mean height. We then use RANSAC and a regressor to fit a plane to the points; the height for each point along the boundary is then calculated using the estimated ground plane. The road points are projected to the image and a depth mask is created, obtaining the amodal label for the road. The ground planes can be projected to a particular location of a newly collected route by matching the location to the labeled map with GPS and refining alignment using ICP. A final check for the ICP solution is made by verifying that the mean projected groundtruth mask for the road label meets an 80% mean IOU with all the other groundtruth masks for the same location, if not the query location data is not retrieved. Ground truth road masks and a labeling example are shown in Appendix A.1.

3.5. Amodal Instance Segmentations: Objects

Amodal objects were labeled using Scale AI for six foreground object classes: car, bus, truck (includes cargo, fire truck, pickup, ambulance), pedestrians, cyclist and motorcyclist. There are three main components in this labeling paradigm: first identifying the visible instances of objects, then inferring the occluded instance segmentation mask, and finally labeling the occlusion order for each object. The labeling is performed on the leftmost front facing camera view. We follow the same standard as KINS [28]. We have labeled a set of 7000 images across various weathers and locations with split details in Section 5.2. Example ground truth amodal masks are shown in Figure 5.

4. Algorithm for Amodal Road Segmentation

To showcase the environmental diversity and amodal qualities of the dataset, we trained and tested two baseline networks to identify the amodal road at pixel levels, even if the road is covered by snow or cars. The first baseline network is **Semantic Foreground Inpainting (SFI)** [23]. We show some qualitative results of this model under night, snow, and cluttered environments in Appendix D. This network fails in conditions with low-light, extreme weather, long-distances, and cluttered pixels. We therefore propose a second baseline which improves upon SFI through the following three innovations in Figure 6.

Positional and Channel Attention: Because amodal segmentation is primarily about inferring what is *not* visible,

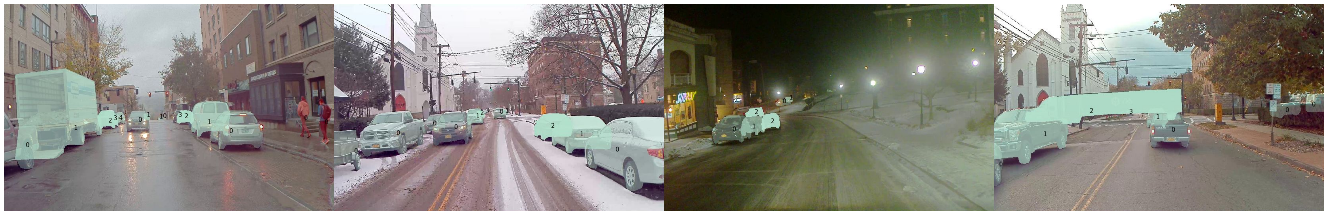


Figure 5. Examples of amodal ground truth masks for cars across different weather conditions.

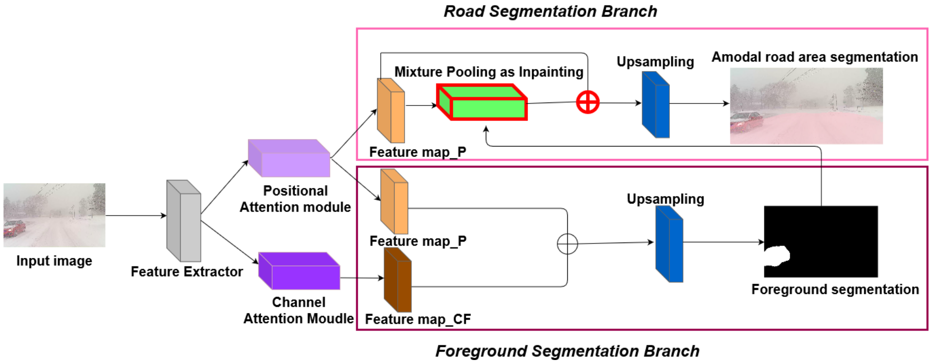


Figure 6. Our proposed amodal road segmentation baseline using PAM, CAM, and modified pooling modules, designed to restore occluded background features. Bold red indicates that the Mixture Pooling as an Inpainting module is added to the road segmentation branch.

context is an extremely important cue. Fu et al. [12] introduce two innovations to capture two different kinds of context. The positional attention module (PAM) uses pixel features to attend to other pixels in the image, in effect capturing context from other parts of the image. The channel attention module uses a similar attention mechanism, effectively aggregating information over *channels*. We apply both these modules over the backbone feature extractor as shown in Figure 6. (Details of these modules are in Appendix B). We combine CAM and PAM to better locate the fine mask boundary. The final foreground instance mask is obtained by an up-sampling layer.

Mixture Pooling as Inpainting: The Max Pooling as Inpainting operation proposed by [23] can be used to help restore the amodal road features by replacing overlapping foreground features with nearby background features. However, since the background features are usually smoothly distributed, a max pooling operation will be very sensitive to any noise added to them. In contrast, an average pooling operation could naturally mitigate the noise. To this end, we propose to combine average pooling and max pooling for inpainting, which we name Mixture Pooling. We show some qualitative results in Appendix B: the Mixture Pooling module produces foreground masks of smoother boundaries than the original max pooling operation.

Sum operation: Before the final up-sampling layer, instead of directly passing the features from the mixture pooling module, we include a residual link from the output of the PAM module. By optimizing the two feature maps jointly in the road segmentation branch, the PAM module can also learn the background features in occluded regions. This can lead to a more accurate restoration of the background features. We include qualitative results in Appendix B to demonstrate that the sum operation produces better inpainting results than the network's performance without the sum operation, especially when the foreground objects (*e.g.*, cars) occupy a relatively large space in the images.

5. Experimental Baselines and Results

5.1. Amodal Road Segmentation

Dataset splits for road. One of the main goals of this dataset is to assess the performance of amodal road segmentation across weather conditions. To create a training, validation, and test set, we pick 590 locations throughout the route and retrieved images and road depth masks for each of these locations for 25 different traversals. After filtering out locations with poor alignment after ICP (as explained in the labeling section), we have a dataset with 11,475 images for training and 3,275 for testing. *We ensure that the locations in testing are not visible from the locations in training.* Additionally, we further split this dataset by weather condition, creating 5 smaller datasets across all weather conditions (sunny, rainy, cloudy, snowy, and night). For each condition, we include the same number of images for each location. These smaller datasets are meant to study the effect of domain gaps between weather conditions, given that all datasets have the same amount of images/data and have seen the same locations. Details of the number of images in our dataset along with other published test sets from KITTI

Table 1. Dataset splits (# of images) of amodal road segmentation.

Dataset	Train Set	Test Set
Cityscapes [9]	2,975	500
KITTI [10]	N/A	200
Ithaca365 (All)	11,475	3,275
Ithaca365 (each weather)	2,295	655

Table 2. **Amodal road results (IOU for road) among five weather conditions.** We report training on *row* and testing on *column*, using the SFI/OURS models. Diagonal entries are the in-domain model. The best result for each column is **bold**.

Far	Sunny	Cloudy	Rainy	Snowy	Night
Sunny	**49.0/ 57.3**	44.0/ 51.7	44.6/ 49.8	40.6/ 47.0	22.0/ 46.7
Cloudy	37.8/ 48.8	**45.8/ 58.0**	37.1/ 49.1	36.2/ 44.1	26.0/ 31.2
Rainy	39.0/ 46.6	45.1/ 53.8	**46.5/ 55.5**	37.9/ 46.3	29.7/ 45.0
Snowy	40.5/ 44.9	41.4/ 52.0	38.8/ 49.6	**44.0/ 54.6**	35.4/ 46.6
Night	31.1/ 43.7	36.6/ 50.3	33.9/ 48.5	30.4/ 52.4	**38.2/ 55.1**
Close	Sunny	Cloudy	Rainy	Snowy	Night
Sunny	**91.6/95.5**	89.4/ 92.0	86.5/ 88.4	85.7/ 88.7	82.8/ 90.0
Cloudy	91.2/ 85.6	**91.9/ 95.7**	83.0/ 86.8	83.4/ 88.5	82.9/ 86.8
Rainy	89.1/ 90.7	87.7/ 89.0	**89.7/ 94.9**	85.6/ 88.3	85.3/ 89.2
Snowy	86.8/ 89.0	90.7/ 92.2	84.3/ 86.5	**91.2/ 92.9**	89.3/ 91.5
Night	87.5/ 90.3	90.5/ 93.6	85.4/ 87.2	89.4/ 92.0	**91.3/ 93.6**

road [10] and Cityscapes [9] are shown in Table 1.

Evaluation metrics for road. Segmentation tasks are typically evaluated through mean IOU. However, we find that this metric is not appropriate for amodal road segmentation. The vast majority of the road pixels in an image correspond to the first few meters in front of the car, which are easy to segment. We find that our baselines often achieve a high IOU but only correctly segment the immediate vicinity of the ego-car (Figure 7). However, for self-driving, accurate segmentation of far-away pixels is needed, as they influence planning and future decisions.

To ensure that algorithms do not ignore far-away road pixels, we introduce a metric that divides the evaluation into two bins, *close* and *far*, by using the depth mask. We denote a range of 30m or less as *close* and anything above as *far*. This follows a similar custom to other datasets such as KITTI, that bin object detection, as *easy*, *medium*, or *hard* based on distance. In Table 2 and Figure 7, we quantitatively and qualitatively demonstrate the need for an evaluation that is based on distance. Quantitatively, we can see that the model performance is high and does not vary much when looking at close ranges and even across weather conditions. Qualitatively, in Figure 7, we can see that below the threshold distance the model captures the road well, but above it, branches or far away roads are not captured.

Experimental results. We perform experiments on our dataset, along with the publicly released training/test sets for CityScapes and KITTI [23]. Before the two baseline models (see Section 4) are trained using our amodal road dataset, we generate the required foreground semantic masks (*i.e.* cars, trucks, buses, and persons) for both baselines using a pretrained DeepLab [7] network. Our baseline model is trained using two GPUs (NVIDIA GT 1080 Ti) in parallel and the SFI is trained using one. Both of the networks are trained using Adam optimizer with a batch size of 8. The overall experimental results are shown in Table 3.

Table 3. Amodal road segmentation results (mIOU) for SFI/OURS models trained/tested on Cityscapes, KITTI, and Ithaca365.

Model	Train Set	IOU on Test Set		
		KITTI [10]	Cityscapes [9]	Ours
SFI [23]	Cityscapes	66.29	72.75	50.16
SFI [23]	Ithaca365	**72.31**	**74.68**	**89.50**
Ours	Ithaca365	72.25	77.43	92.19

We observe that models trained on our dataset are in general better than those trained on Cityscapes. For instance, when tested on ours, the SFI model trained on ours significantly outperforms that trained on Cityspaces (89.50 vs. 50.16). While this may not be surprising, as in-domain results are normally better than out-domain results, reversely when tested on Cityspaces, the SFI model trained on ours is still better than that trained on Cityspaces (74.68 vs. 72.75). Moreover, when tested on KITTI, the SFI model trained on ours notably outperforms that trained on Cityspaces. These results together suggest that our dataset is more useful for training a robust and generalizable model. Also, we want to note that the 2,975 Cityscapes images used in SFI [24] have normal modal segmentation masks thus they are weak labels. Additionally for the test set, the authors of SFI manually labeled 500 images with the amodal masks for three background classes (*sidewalk, road, other rigid world*). Due to the difficulty of annotating amodal masks based on one image especially for far away and occluded pixels, the annotations are coarse and are prone to errors. We believe weak labels and noisy ground truths lead to the large gap in model performance between the Cityscapes and Ithaca365. Across models (SFI vs. ours), we find that our proposed model is better than prior art.

We also evaluate the SFI and our baseline models across 5 different weather conditions (Table 2) by using our proposed Far and Close IOU metrics . In close quarters, both methods perform well, yet large performance gaps are found for pixels farther than the 30m threshold — either between SFI and our models, or between in-domain and cross-domain results. This indicates that 1) there are significant domain gaps between different weather/time conditions, and revealing these gaps requires stratifying evaluations by distance; 2) our model is more robust than the SFI model. Snowy and night appear to be the most challenging weather conditions for this task since the models that are not trained under these conditions perform worse under these two conditions. One potential reason for this phenom-

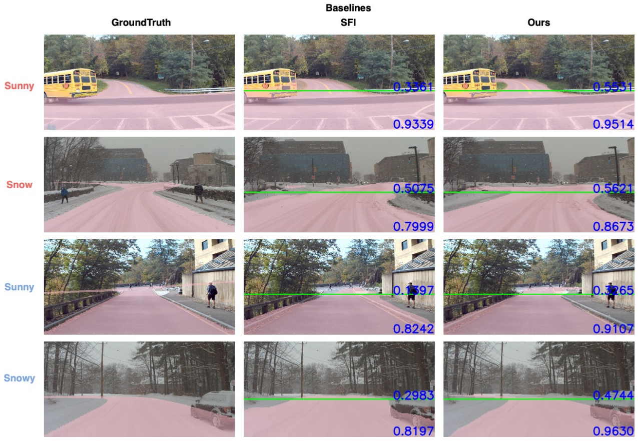


Figure 7. Amodal road segmentation results for the two baselines (SFI, OURS). Top two rows are trained on sunny conditions, with the first row tested on sunny and second on snowy. Bottom two rows are trained on snowy conditions, with the third row tested on sunny and the fourth on snowy. The cutoff depth of 30m is marked with a green line, with the *far* IOU above and *close* IOU below.

ena is that snowy and night have the most visual degradation among all weather conditions, which cause obstacles during the feature extraction stage for both models.

5.2. Amodal Instance Segmentation

Our second set of experiments is on amodal instance segmentation, using our annotated 7,000 images of six classes. We adopt Mask R-CNN [15, 16] as the baseline model. We conduct two experiments. First, we train the model using our overall amodal training set, which consists of 5,600 images collected across five different weather conditions. We then evaluate the model on our overall amodal test set, which consists of 1,400 images (Table 4). The training and test data are strictly from different locations. We achieve a decent overall mean average precision (mAP) of 56.5.

Second, we split the dataset into five different weather conditions (sunny, snowy, cloudy, rainy and night). On each of the five weather conditions we prepare train and test sets with different quantities of images as different weathers vary in frequency across the 40 labeled traversals. In general, the train set has 1120 images and the test set has 280 images but rainy and snow conditions have less. Then for each weather condition, we train Mask R-CNN models respectively and evaluate the mAP of the trained model on the test set of all five weather conditions. As shown in Table 4, for all five weather conditions, training and testing on the same weather condition has the highest mAP. This confirms that that domain adaption is a challenging task for Amodal instance Segmentation. We still observe that snowy, night and rainy have lower mAP than sunny and cloudy, meaning that they are more challenging cases and deserve further, more focused research efforts. Visualizations of the amodal instance segmentation are included in Appendix D.

Table 4. **Amodal instance segmentation results (mAP@[.5:.95]) across five weather conditions, using Mask R-CNN [16]**. Each entry means training on *row* and testing on *column*. The best result for each column is **bold**.

Train \Test	Sunny	Cloudy	Rainy	Snowy	Night
Sunny	**54.3**	47.6	38.9	41.1	29.4
Cloudy	47.2	**48.0**	33.6	21.4	14.2
Rainy	35.5	32.1	**46.3**	33.2	21.0
Snowy	33.9	40.4	35.8	**47.0**	21.4
Night	26.7	27.7	24.9	23.7	**30.2**

Table 5. **3D car detection result on our dataset, using pre-trained Point R-CNN [31] on KITTI and Argoverse**. We report AP_{BEV}/AP_{3D} of the car category at IoU= 0.5/0.7 across different depth ranges on the test set.

	IoU 0.5			IoU 0.7		
Model	0-30	30-50	50-80	0-30	30-50	50-80
KITTI	64.2 / 61.8	45.3 / 42.7	26.3 / 20.8	41.5 / 26.0	32.3 / 13.1	10.2 / 5.3
Argo	54.2 / 53.1	33.6 / 27.5	15.4 / 13.8	40.2 / 15.9	18.3 / 6.6	13.3 / 9.1

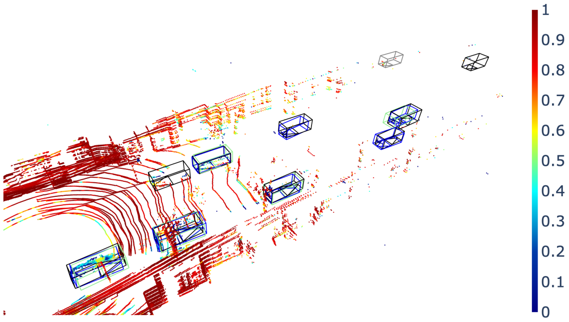


Figure 8. **Ephemerality from multiple traversals.** With observations from multiple traversals, we compute the *ephemerality* statistics introduced in [2], and color the point cloud by the scale. We show ground-truth labels in black, true positive detections from models trained on KITTI in green, on Argoverse in blue, and false positive ones in gray. Multiple traversals enable easy false positive filtering by thresholding the ephemerality.

5.3. 3D Object Detection, Repeated Routes

Our third set of experiments is on 3D object detection using LiDAR data. For this experiment a small test set consisting of 175 point clouds at different locations, and annotated 3D bounding boxes for six classes (car, truck, bus, cyclist, pedestrian) is used. We then evaluate a LiDAR detector, Point-RCNN [31], pre-trained on KITTI and Argoverse. Point-RCNN is a popular method which incorporates PointNet [27] point wise features and faster R-CNN [30] to directly generate proposals in 3D using only LiDAR points.

We follow KITTI to report average precision (AP) in 3D and bird-eye-view (BEV), denoted by AP_{3D} and AP_{BEV}, respectively. We split AP by depth ranges. We focus on the *Car* category as it is the most commonly observed class. We report *Car* AP with the intersection over union (IoU) thresholds at 0.5 or 0.7, meaning, a ground truth car is correctly detected if the IoU between it and the detected box is larger than 0.5 or 0.7. The result is shown in Table 5. This

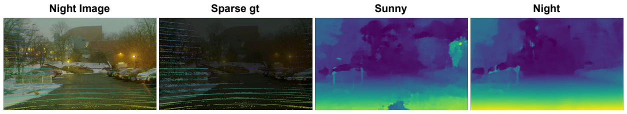


Figure 9. Qualitative results of disparity maps, using PSMNet [5] trained on different weather conditions.

Table 6. **Three-pixel-error (%) of stereo matching task using PSMNet [5] among four different conditions.** On each entry (*row*, *column*), we report training on *row* and testing on *column*.

Train \Test	Sunny	Rain	Snow	Night
Sunny	9.48	23.58	24.41	17.79
Rain	12.42	24.05	27.45	19.90
Snow	13.01	27.84	20.99	22.83
Night	10.58	23.6	24.46	11.99

serves as a baseline to further study the use of our dataset for 3D object detection across weather conditions and repeated routes. We further demonstrate the *ephemerality* statistics introduced in [2], which make use of multiple traversals of the same location to determine if a point is ephemeral, i.e., not persistent (see Figure 8). This information can be used to remove false positives to improve a detector's mAP, if we have traversed the same location multiple times earlier.

5.4. Stereo Disparity Estimation

Our final experiment is on stereo disparity estimation to further showcase the challenge of diverse weather and lighting conditions of our dataset. For each of the conditions (sunny, snow, rain, and night), we collect 4,739/1,188 left-right image pairs (the baseline is 0.46 m) for training and testing. The LiDAR data used as depth ground truth in this experiment is from a previous sensor setup where 4 VLP-16s were mounted in front of the vehicle but the cameras remain unchanged. We obtain sparse ground-truth depths (and so pixel disparities) for pixels that have corresponding LiDAR points. We then train and test a baseline model of stereo matching PSMNet [5] on each condition.

Table 6 shows the results. We again observe a large domain gap, even in the low-level vision task of stereo matching: the diagonal in-domain errors are usually much smaller than other cross-domain errors of the same columns. This can also be seen qualitatively in Figure 9: on a night-time image, PSMNet trained using night images notably outperforms using sunny images. Besides, by comparing different columns (different testing cases), we clearly see the difficulty in performing stereo disparity matching at rainy, snowy, and night conditions than the sunny condition. All these results again suggest the timely need of a dataset like ours, with more diverse weather conditions.

6. Conclusion, Discussion, Data Release Plan

Table 7. Table of train/test data split we used for amodal road segmentation, instance segmentation and 3D object detection tasks. The first column indicates the total frames in our dataset.

	Total	Train	Test
#frames	682217	N/A	N/A
#frames w/ amodal road masks	14750	11475	3275
#frames w/ amodal instance masks	7000	5600	1400
#pcls w/ 3D bounding boxes	175	N/A	175

Current challenges in self-driving car perception are lagging due to the lack of datasets with poor weather, varying light, and dense traffic conditions. This paper develops a unique dataset for the research community, with amodal road and object mask labels across varying scenes (university campus, downtown, highway, urban, residential, and rural) and environmental conditions (snow, rain, night). By using data across repeated traversals of the same route, we can efficiently create a large scale, diverse dataset with amodal labels. The repeated routes also opens potential new research directions such as object discovery and continual learning. A new metric is introduced for the proper evaluation of amodal segmentation of background classes such as road. Baseline road detection, depth estimation, instance segmentation, and 3D detection using this dataset reveals significant performance gaps under varying weather conditions, demonstrating the usefulness of this dataset.

A key advantage of our dataset is the availability of multiple traversals over the same route. These repeated traversals capture the reality that many people repeatedly drive the same routes between home, work and leisure. Information from past traversals can not only improve detections [1] but also allow one to distinguish between static background and mobile objects in the foreground without supervision [2]. As such, our dataset is a useful testbed for *weakly supervised/unsupervised 3D object discovery and detection*. We argue that research on this problem is the need of the hour: current reliance on costly human-annotated data is limiting.

Data release plan: Ithaca365 contains *aligned* LiDAR point clouds as well as images captured across 40 traversals of our 15km route. We will provide ground-truth 3D object bounding boxes and amodal masks for six object classes as well as ground truth amodal masks on 7K selected frames. These numbers are similar to the KITTI benchmark [14]. The validation data is carefully selected from different geo-locations from the training data, on different days. We will continue growing the dataset even after this initial release by collecting data over future seasons. IRB deemed this project not Human Participant Research therefore does not require IRB approval.

7. Acknowledgements

This research is supported by grants from NSF (IIS-1724282, IIS-2107161, IIS-2107077), ONR (N00014-17-1-2175), and SRC: 2019-AU-2891. We also thank Scale AI, Amazon (Ground Truth Plus), and Point One Navigation.

References

[1] Anonymous. Hindsight is 20/20: Leveraging past traversals to aid 3d perception. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. under review. 8

[2] Dan Barnes, Will Maddern, Geoffrey Pascoe, and Ingmar Posner. Driven to distraction: Self-supervised distractor learning for robust monocular visual odometry in urban environments. In *ICRA*, pages 1894–1900. IEEE, 2018. 7, 8

[3] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 3

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *preprint, arXiv:1903.11027*, 2019. 1, 2

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network, 2018. 1, 8

[6] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps. In *CVPR*, 2019. 1, 2

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 6

[8] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *ICCV*, 2019. 1

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2, 6

[10] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013. 6

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, pages 2002–2011, 2018. 1

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 5, 3

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 2, 8

[15] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron. https://github.com/facebookresearch/detectron, 2018. 7

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2, 7

[17] Radu Horaud and Fadi Dornaika. Hand-eye calibration. *The international journal of robotics research*, 14(3):195–210, 1995. 4

[18] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. url-https://level5.lyft.com/dataset/, 2019. 1, 2

[19] Hendrik Königshof, Niels Ole Salscheider, and Christoph Stiller. Realtime 3d object detection for automated driving using stereo vision and semantic information. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, 2019. 1

[20] Ke Li and Jitendra Malik. Amodal instance segmentation. In *European Conference on Computer Vision*, pages 677–693. Springer, 2016. 1, 3

[21] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2359–2367, 2017. 1

[22] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 1

[23] Chenyang Lu and Gijs Dubbelman. Semantic foreground inpainting from weak supervision. *IEEE Robotics and Automation Letters*, 5(2):1334–1341, 2020. 1, 2, 3, 4, 5, 6

[24] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 1, 2

[25] Teddy Ort, Igor Gilitschenski, , and Daniela Rus. Grounded: The localizing ground penetrating radar evaluation dataset. In *Robotics: Science and Systems*, 2021. 2

[26] Matthew Pitropov, Danson Evan Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Lake Waslander. Canadian adverse driving conditions dataset. *CoRR*, abs/2001.10117, 2020. 1, 2

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 7

[28] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 3, 4

[29] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *CVPR*, 2020. 1

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 7

[31] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 7

[32] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2019. 1, 2

[33] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2

[34] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019. 1

[35] Yan Wang, Zihang Lai, Gao Huang, Brian H Wang, Laurens van der Maaten, Mark Campbell, and Kilian Q Weinberger. Anytime stereo image depth estimation on mobile devices. *preprint, arXiv:1810.11408*, 2018. 1

[36] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *IROS*, 2019. 1

[37] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation, 2021. 2, 3

[38] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Realtime 3d object detection from point clouds. In *CVPR*, 2018. 1

[39] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017. 1, 2

[40] Yan Zhu, Yuandong Tian, Dimitris Metaxas, and Piotr Dollár. Semantic amodal segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1472, 2017. 1

Supplementary Material

Supplementary Material

Appendix

We provide details omitted in the main text.

- Appendix A: Dataset details. Sensor time synchronization (cf. subsection 3.1 of the main paper). Road labeling pipeline and additional visualizations of generated depth mask ground-truth (cf. subsection 3.4 of the main paper). Dataset class statistics.
- Appendix B: Additional details on baseline algorithms (cf. section 4 of the main paper).
- Appendix C: Link to source code for training and inference (cf. subsection 5.1 of the main paper).
- Appendix D: Training details and visualization results (cf. subsection 5.1 and of subsection 5.2 the main paper).

A. Dataset details

A.1. Visualizations of depth mask ground-truth

We show a labeling example in Figure 10 as explained in section subsection 3.4. Additional ground truth road depth masks are shown in Figure 11.

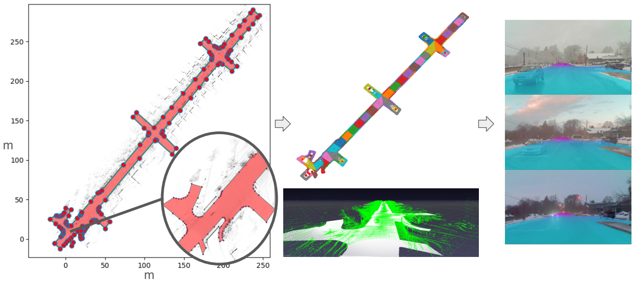


Figure 10. A built pointcloud is projected to BEV and the road is annotated using polygon labeling tool (left). The polygon is then divided into smaller 150 m^2 polygons and ground planes are estimated. Each color in the polygon represents a subdivision (middle). Finally ground planes are projected onto the image yielding amodal road mask with depth. Purple means farther distance (right)

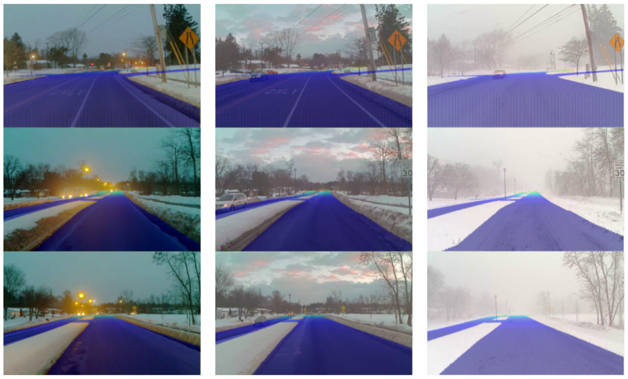


Figure 11. Generated ground truth depth masks for each pixel lying in the road area. Lighter blue indicates farther depth.

A.2. Sensor time synchronization details.

We use NVIDIA's recording tool to log the LiDAR at 10 Hz; cameras at 30 Hz. The Novatel GPS/INS data was logged at 100 Hz using the PC running ROS, time synchronized with the AGX through PTP from the Novatel custom firmware. The timing synchronization among cameras has been verified to average 60 μs. We select the OS2 as the reference to which all other sensors are matched. The average time difference between this LiDAR and the cameras is 8.9 ms, with a worst case of 16.6 ms when there are no camera frames dropped. For the IMU/GPS, the average time difference is 3 ms; the worst case is 30 ms. The INS/GPS poses can be interpolated to the selected LiDAR time. For the VLPs, the worst case time difference is 35 ms.

A.3. Dataset Statistics

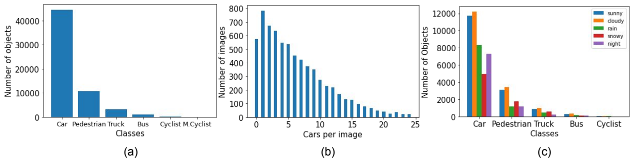


Figure 12. Statistics for a) overall object counts; b) cars per image; c) object distributions over weathers.

We include statistics on the amodal object labels in Figure 12. Regarding the amodal road segmentation task the average number of close pixels per image are 767,759 while far are 38,552, which corresponds to 33.0% and 1.7% of the image, respectively.

B. Baseline Algorithm Details

B.1. Dual Attention network details

As discussed in section 4, we add a positional attention module (PAM) and channel attention module (CAM) to our baseline. In Figure 13 we show diagrams for these two modules.

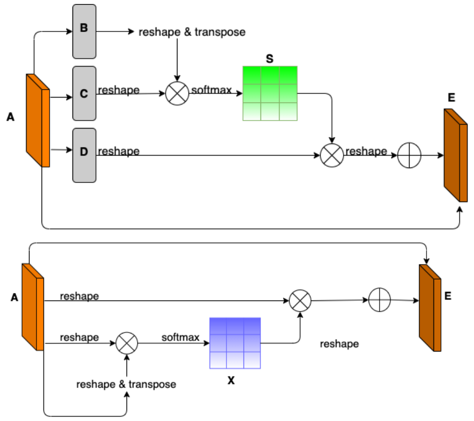


Figure 13. The image here shows the PAM module (top) and the CAM module (bottom) proposed by [12]

B.2. Pooling operation

The detailed algorithm for Mixture Pooling as Inpainting is shown in Algorithm 1, mentioned in section 4. This is a modification from Max Pooling as Inpainting, we refer the reader to [23] for that algorithm. Qualitative results comparing Max Pooling vs Mixture pooling are demonstrated in Figure 14.

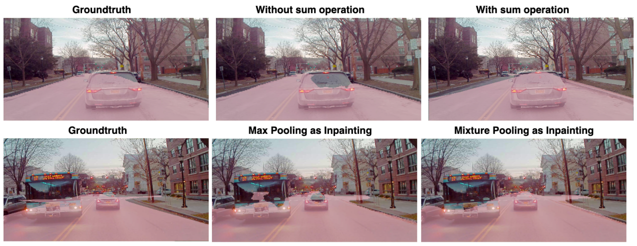


Figure 14. The top row shows our proposed baseline prediction results with Max Pooling as Inpainting and Mixture Pooling as Inpainting respectively. The bottom row shows our proposed baseline prediction results with and without sum operation in the road segmentation branch respectively.

C. Training and inference code: amodal road

Training and inference code is here: https://github.com/coolgrasshopper/amodal_road_segmentation

D. Training details & visualization results

D.1. Hyperparameters & Error analysis

The detailed hyperparameters for our baseline network discussed in section 4 are demonstrated in our released source code. In general, we train the network using 240 epochs with initial learning rate 0.3 and weight decay $1e-$04 through the training process. Additionally, we set different seeds three times and record the error bars for our proposed baseline in Table 9. Also, the standard deviation for the three conducted experiments are 0.042% and 0.041% for the *far* and *close* IOU respectively. For both the *close* and *far* IOU, the errors and standard deviations are within 0.1% using our proposed baseline. This shows that our experiments' results are reproducible.

Algorithm 1 Mixture Pooling as Inpainting

```
Require: $F_{raw}$ - intermediate feature map; $M$ binary foreground mask with 1 being background
Ensure: $F_{inpainted}$ is inpainted feature map
  $M = \text{max-pooling}(M)$
  $F_{background} = F_{raw} \times M$
  $F_{patch}$ = zero tensor with size same as $F_{background}$
  $M_{old} = M$
  do
      $F_{background} = 0.5*$ max-pooling $(F_{background})$ $+0.5*$ avg-pooling $(F_{background})$
      $M_{new} =$ max-pooling $(M_{old})$
      $F_{patch} += (M_{new} - M_{old}) * F_{background}$
      $M_{old} = M_{new}$
  while 0 still exists in $M_{old}$
  $F_{inpainted} = F_{raw} * M + F_{patch}$
```

D.2. Qualitative performance evaluation of SFI

The qualitative evaluation of SFI is shown in Figure 15 demonstrating the failure cases of SFI under challenging scenarios (*e.g.* cluttered scenes, night and faraway regions).

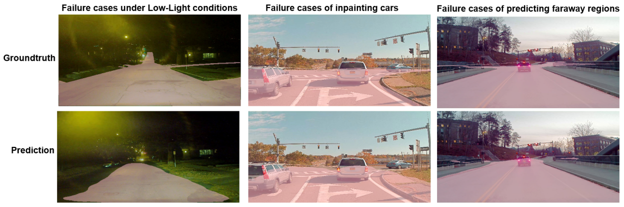

Failure cases under Low-Light conditions
Failure cases of inpainting cars
Failure cases of predicting faraway regions
Groundtruth
Prediction

Figure 15. Visualized failure cases of the Semantic Foreground Inpainting network. Ground-truth overlay results are shown in the top row and network prediction results are in the bottom row.

D.3. *Far* & *Close* IOU calculation

To calculate the *far* and *close* IOU, for each image k, we group each pixel i based on its the depth d_{ki}. The set which contains all pixels with $d_{ki} < 30$ is denoted as DC_k. The set that contains all pixels with $d_{ki} \geq 30$ is denoted as DF_k. After that, for the test dataset that contains N total test images, the *close* IOU is calculated as:

$$\frac{\sum_{k=0}^{N} I_{DC_{kg}} \cap I_{DC_{kp}}}{\sum_{k=0}^{N} I_{DC_{kg}} \cup I_{DC_{kp}}} \tag{1}$$

where $I_{DC_{kg}}$ demonstrates for pixels with depth $d_{ki} < 30$, the ground-truth binary mask for Amodal road segmentation at image k and $I_{DC_{kp}}$ indicates the Amodal road segmentation prediction at image k for *close* pixels.

Similarly, the *far* IOU is calcuated as:

$$\frac{\sum_{k=0}^{N} I_{DF_{kg}} \cap I_{DF_{kp}}}{\sum_{k=0}^{N} I_{DF_{kg}} \cup I_{DF_{kp}}} \tag{2}$$

where $I_{DF_{kg}}$ demonstrates for pixels with depth $d_{ki} \geq 30$, the ground-truth binary mask for Amodal road segmentation at image k and $I_{DF_{kp}}$ indicates the Amodal road segmentation prediction at image k for *far* pixels.

We also attached the evaluation code in our anonymously released code: https://github.com/coolgrasshopper/amodal_road_segmentation

D.4. Split by location results

To investigate the effects of training on different road types and surrounding environments, we split the dataset into five different areas: urban, highway, rural, campus, downtown. Then, we train **our** proposed baseline using images collected at each location respectively. For each trained model, we test the model's performance using the dataset collected under the other four location types. The detailed performance is illustrated in Table 8. We observe that many models seem to drop in performance in the campus and urban environment. A potential reason for this is that urban areas contain more curved roads, and higher levels of occlusions and less visibility due to nearby buildings and vehicles, while campus contains some complex road structures with intersections and road islands (islets).

D.5. Ablation study of our proposed baseline

We conduct an ablation study of our proposed model by removing the added sum operation in the road segmentation branch and not modifying the original Max pooling as Inpainting operation (*i.e.*, removing the mixture pooling operation). That is, we remove the sum operation but keep Mixture pooling as Inpainting (third row of Table 9). We also keep the sum operation but use Max pooling as in painting (fourth row of Table 9). Finally we also add the feature map from the channel attention module (fifth row of Table 9). This channel attention module is to stress the inter-dependencies of feature maps. As each feature map can be regarded as a class-specific response, channel attention can also be interpreted as emphasizing the inter-dependencies of different (foreground) classes. For road segmentation, since we only have one semantic class, we remove the module to save computation in our proposed baseline. We test all trained models using the total collected test dataset and illustrate the *close* and *far* IOU. The results are shown Table 9. We also demonstrate the *close* and *far* IOU performances of SFI and our proposed baselines in Table 9.

Table 8. **Results (IOU for road) on model performance among five different locations.** On each entry (*row*, *column*), we report training on *row* and testing on *column* using our proposed baseline.

Far IOU	Urban	Downtown	Highway	Rural	Campus
Urban	N/A	46.93	38.57	43.40	35.92
Downtown	51.43	N/A	51.93	55.81	41.55
Highway	36.76	46.11	N/A	57.35	39.19
Rural	43.64	47.18	53.92	N/A	39.73
Campus	42.53	50.55	53.63	57.16	N/A
Close IOU	**Urban**	**Downtown**	**Highway**	**Rural**	**Campus**
Urban	N/A	94.04	88.11	87.54	92.16
Downtown	90.19	N/A	94.17	88.46	93.79
Highway	86.54	92.90	N/A	88.27	91.36
Rural	92.89	93.40	91.95	N/A	91.17
Campus	92.95	94.32	94.21	89.94	N/A

Table 9. **Ablation study** that removes the sum operation ('w/o sum' row in the table) and the mixture pooling operation ('w/o mix pooling' row in the table) in the road segmentation branch respectively. We also include the *close* and *far* IOU for SFI and our proposed baselines in the table.

Architectures	*Close* IOU	*Far* IOU
SFI	91.55	52.16
w/o sum	93.19	55.06
w/o mix pooling	93.08	54.77
w/ map_{CF}	93.58	57.06
Ours	93.29 (± 0.072)	56.67 (± 0.075)

D.6. Qualitative experiment results

Finally, we demonstrate the visualization results of amodal instance segmentation in Figure 16. From the visualization results, we find that cars and pedestrians are predicted more accurately under sunny situation than snowy and night situations. This underscores visual challenges for amodal scene reasoning under more adverse conditions. Qualitative results split by sunny, rainy, cloudy, night and snowy conditions are shown in Figure 17 to Figure 21 for amodal road segmentation. The green line demonstrates the **closest** horizontal line (height) to the bottom of the image which has depth $d_h \geq 30$. See Section D.3 for detailed calculation).

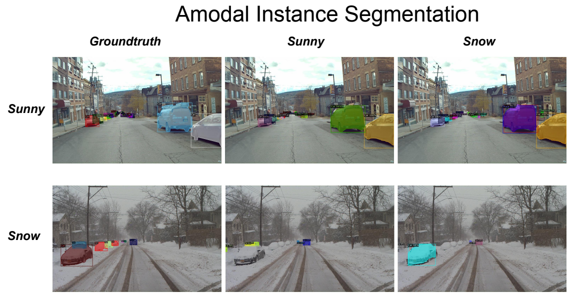


Figure 16. MaskRCNN model trained on the sunny (second column) and snowy (third column) datasets. The groundtruth is shown in the first column. The row indicates the condition being tested on (i.e. sunny first row, snowy second row).

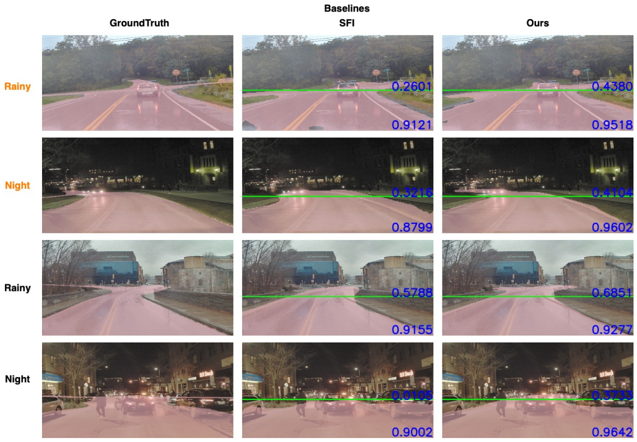


Figure 17. Road inference for the two baselines. Orange indicates models trained on rainy with the first row testing on rainy and second row on night. Black indicates training on night with the third row testing on rainy and the fourth row testing on night. Above green line (30m) is the *far* IOU and below it is *close* IOU.

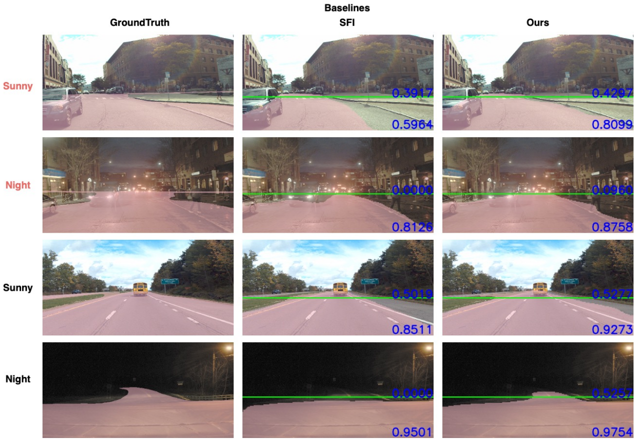


Figure 18. Road inference for two baselines. Red indicates model trained on Sunny with the first row is testing on sunny and second row on night. Black indicates training on night with the third row testing on sunny and the fourth row testing on night. Above green line (30m) is the *far* IOU and below it is *close* IOU.

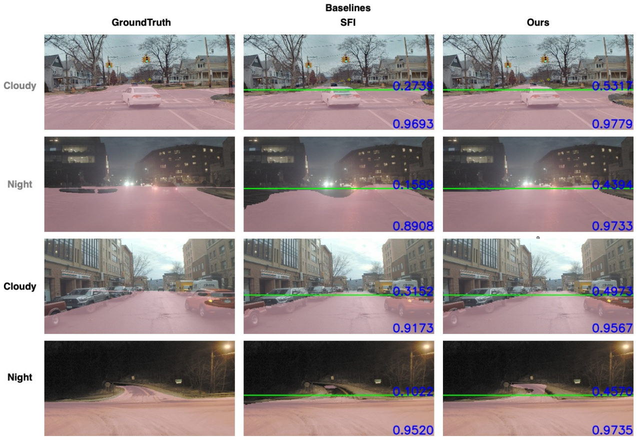


Figure 19. Road inference for two baselines. Grey indicates model trained on cloudy with the first row testing on cloudy and second row on night. Black indicates training on night with the third row testing on cloudy and the fourth row testing on night. Above green line (30m) is the *far* IOU and below it is *close* IOU.

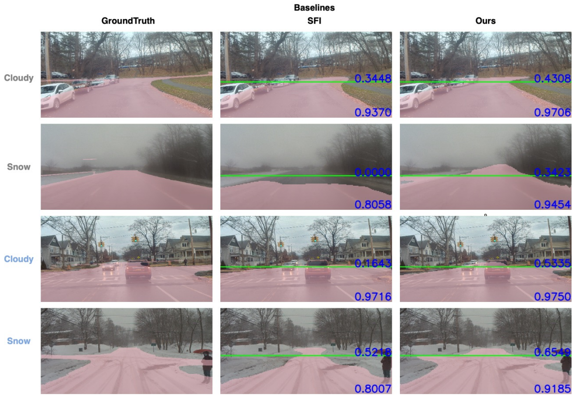


Figure 20. Road inference for two baselines. Grey indicates model trained on cloudy with the first row testing on cloudy and second row on snow. Blue indicates training on snow with the third row testing on cloudy and the fourth row testing on snow. Above green line (30m) is the *far* IOU and below it is *close* IOU.

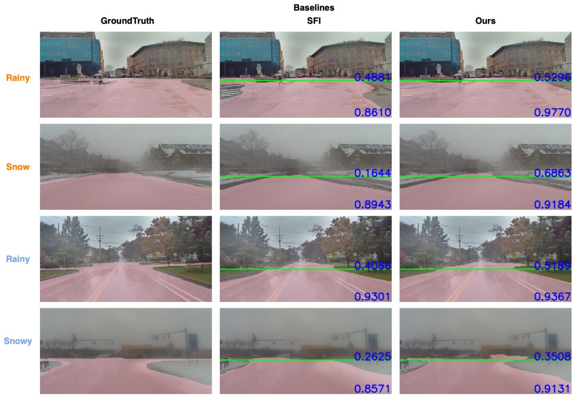


Figure 21. Road inference for two baselines. Orange indicates model trained on rainy with the first row testing on rainy and second row on snow. Blue indicates training on snow with the third row testing on rainy and the fourth row testing on snow. Above green line (30m) is the *far* IOU and below it is *close* IOU.