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.

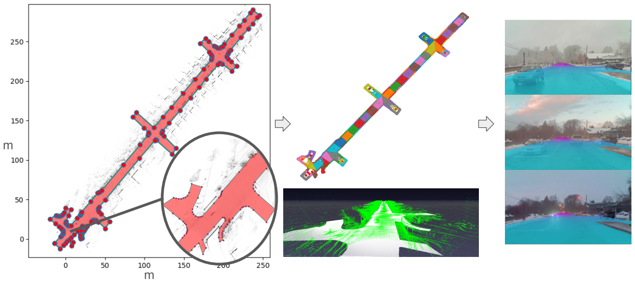

Figure 10. A built pointcloud is projected to BEV and the road is annotated using polygon labeling tool (left). The polygon is then divided into smaller 150 m^2 polygons and ground planes are estimated. Each color in the polygon represents a subdivision (middle). Finally ground planes are projected onto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

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.

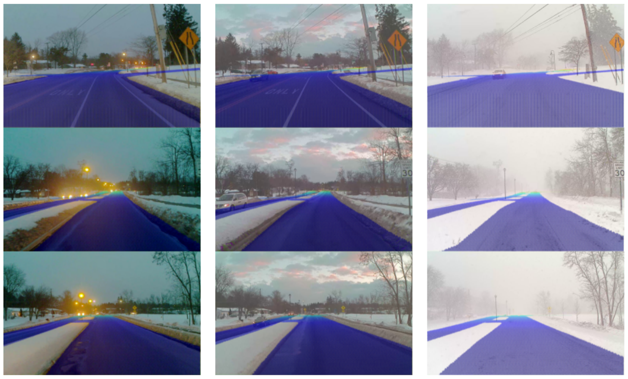

Figure 11. Generated ground truth depth masks for each pixel lying in the road area. Lighter blue indicates farther depth.

A.3. Dataset Statistics

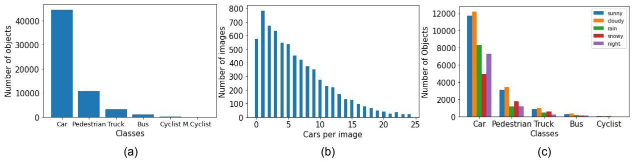

Figure 12. Statistics for a) overall object counts; b) cars per image; c) object distributions over weathers.

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.

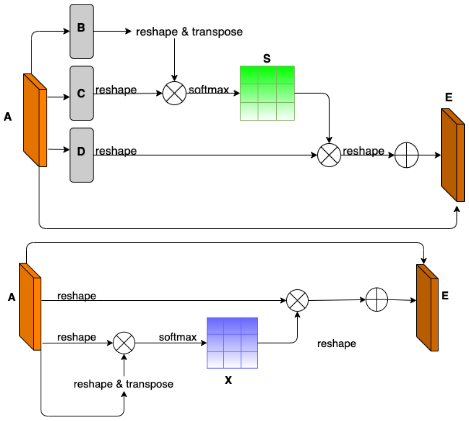

Figure 13. The image here shows the PAM module (top) and the CAM module (bottom) proposed by [12]

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.

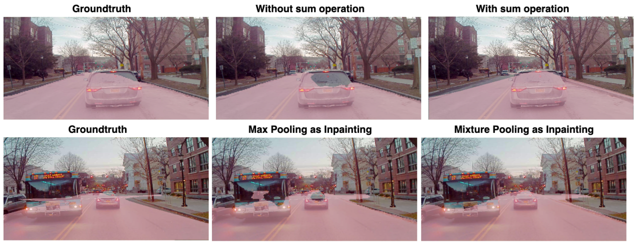

Figure 14. The top row shows our proposed baseline prediction results with Max Pooling as Inpainting and Mixture Pooling as Inpainting respectively. The bottom row shows our proposed baseline prediction results with and without sum operation in the road segmentation branch respectively.

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
  $M = \text{max-pooling}(M)$
  $F_{background} = F_{raw} \times M$
  $F_{patch} = \text{zero tensor with size same as } F_{background}$
  $M_{old} = M$
  do
      $F_{background} = 0.5* \textbf{max-pooling } (F_{background}) +0.5* \textbf{avg-pooling } (F_{background})$
      $M_{new} = \text{max-pooling } (M_{old})$
      $F_{patch} += (M_{new} - M_{old}) * F_{background}$
      $M_{old} = M_{new}$
  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).

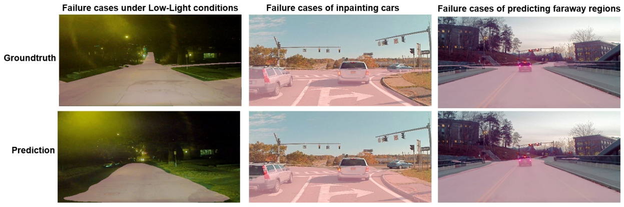
Failure cases under Low-Light conditions
Failure cases of inpainting cars
Failure cases of predicting faraway regions
Groundtruth
Prediction

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 18. 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 19 to Figure 23 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).

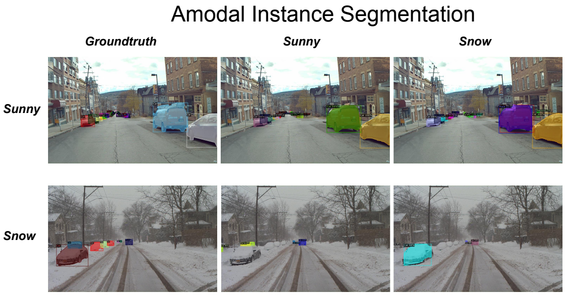

Figure 18. MaskRCNN model trained on the sunny (second column) and snowy (third column) datasets. The groundtruth is shown in the first column. The row indicates the condition being tested on (i.e. sunny first row, snowy second row).

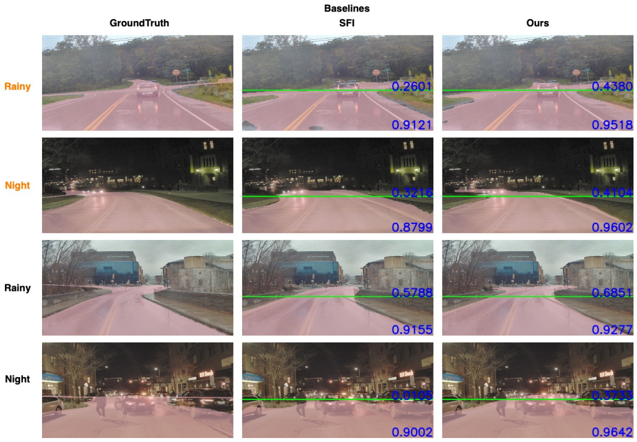

Figure 19. Road inference for the two baselines. Orange indicates models trained on rainy with the first row testing on rainy and second row on night. Black indicates training on night with the third row testing on rainy and the fourth row testing on night. Above green line (30m) is the *far* IOU and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

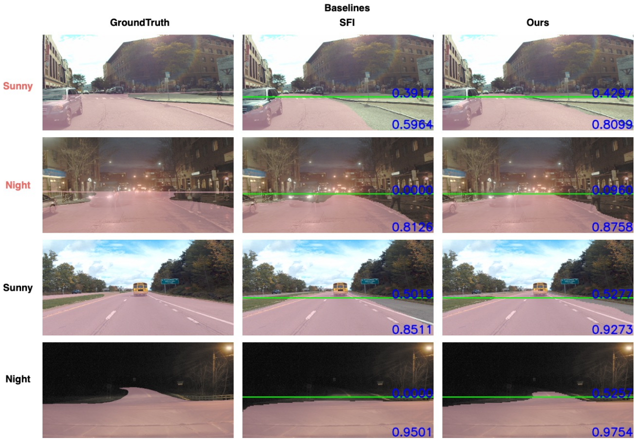

Figure 20. Road inference for two baselines. Red indicates model trained on Sunny with the first row is testing on sunny and second row on night. Black indicates training on night with the third row testing on sunny and the fourth row testing on night. Above green line (30m) is the *far* IOU and below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

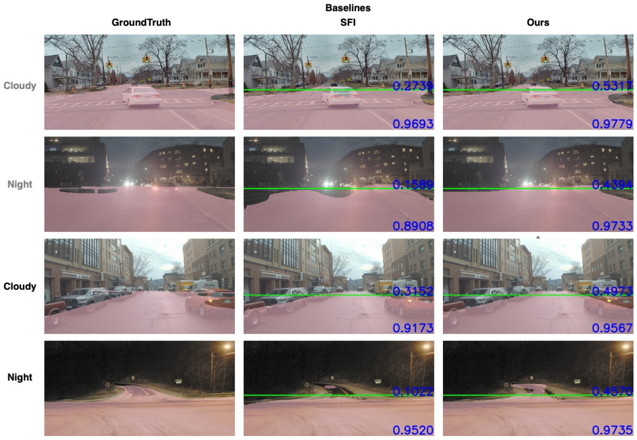

Figure 21. Road inference for two baselines. Grey indicates model trained on cloudy with the first row testing on cloudy and second row on night. Black indicates training on night with the third row testing on cloudy and the fourth row testing on night. Above green line (30m) is the *far* IOU and below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

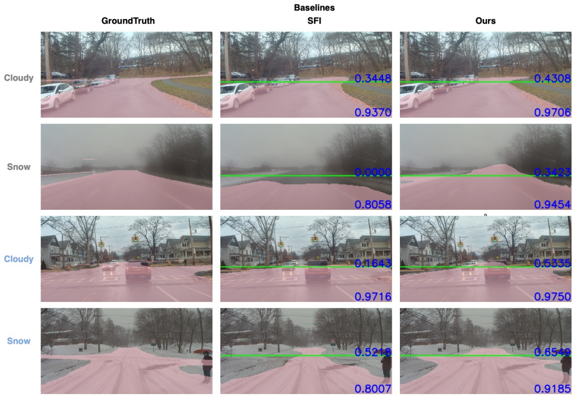

Figure 22. Road inference for two baselines. Grey indicates model trained on cloudy with the first row testing on cloudy and second row on snow. Blue indicates training on snow with the third row testing on cloudy and the fourth row testing on snow. Above green line (30m) is the *far* IOU and below it is *close* IOU.

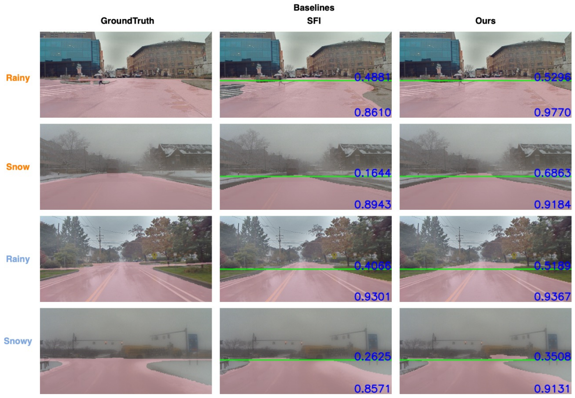

Figure 23. Road inference for two baselines. Orange indicates model trained on rainy with the first row testing on rainy and second row on snow. Blue indicates training on snow with the third row testing on rainy and the fourth row testing on snow. Above green line (30m) is the *far* IOU and below it is *close* IOU.