Amodal Panoptic Segmentation
Supplementary Material

	Simplicity		Convexity	
	Inmodal	Amodal	Inmodal	Amodal
COCO-A [15]	0.746	0.856	0.658	0.685
KINS [10]	0.709	0.830	0.610	0.639
KITTI-360-APS	0.778	0.884	0.689	0.746
BDD100K-APS	0.697	0.821	0.594	0.618

Table 1. Comparison of shape statistics between inmodal and amodal segments in our proposed KITTI-360-APS and BDD100K datasets, along with COCO-A and KINS datasets.

In this supplementary material, we provide additional details on various aspects of our work. We present dataset statistics for our proposed amodal panoptic segmentation datasets in Sec. 1. We then discuss the baseline architectures and the inference in-depth in Sec. 2 and Sec. 3, respectively. Subsequently, we provide details on the loss functions that we employ to train the amodal instance segmentation head of our APSNet in Sec. 4. Finally, we discuss the benchmarking results on the KITTI-360-APS dataset in detail to reinforce the utility of our proposed evaluation metrics in Sec. 5.

1. Dataset

In this section, we present statistics and examples for each of the datasets that we introduce. To evaluate the shape complexity of the amodal segments, we compute the shape convexity and simplicity [15] for each dataset as follows:

$$convexity(S) = \frac{Area(S))}{Area(ConvexHull(S))} \quad (1)$$

$$simplicity(S) = \frac{\sqrt{4\pi * Area(S)}}{Perimeter(S)} \quad (2)$$

Tab. 1 presents the shape complexity metric scores for KITTI-360-APS and BDD-APS datasets. Additionally, we compare the convexity and simplicity of our dataset with existing amodal instance segmentation datasets namely COCO-A [15] and KINS [10].

1.1. KITTI-360-APS

The KITTI-360-APS dataset consists of 11 *stuff* classes namely road, sidewalk, building, wall, fence, pole, traffic sign, vegetation, terrain, and sky. The dataset further comprises 7 *thing* classes, namely car, pedestrians, cyclists, two-wheeler, van, truck, and other vehicles. Tab. 2 presents the *thing* class distribution for the dataset. We observe that the instances of the car are predominant in *thing* classes followed by pedestrian and truck classes. The contribution of the Other-Vehicle class to the number of instances is the least with 0.2%. Fig. 1 (a) illustrates the histogram of occlusion level which is defined as the fraction of occluded region area. We notice about 60% of the instances are either slightly occluded or not occluded at all in the dataset and the rest of the instances have different degrees of occlusions. The second peak in the graph is observed for near moderate occlusion levels while heavily occluded regions are relatively small in comparison. In terms of shape complexity (Tab. 1), KITTI-360-APS consists of relatively simpler amodal segments indicated by the higher the convexity-simplicity average value which is in line with the intuition [15] that independent of scene geometry and occlusion patterns, amodal segments tend to be relatively simpler. Fig. 2 presents examples from our dataset.

1.2. BDD100K-APS

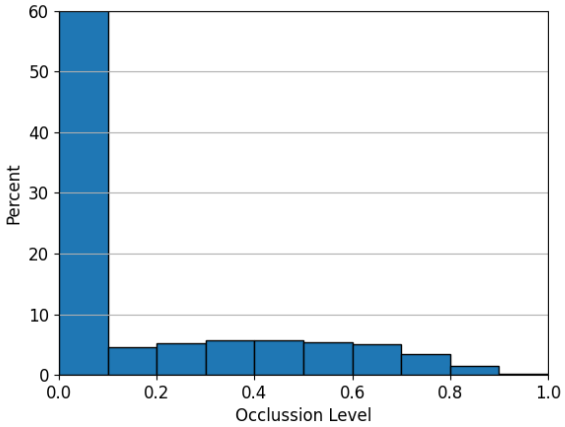
The BDD100K-APS dataset provides amodal panoptic annotations for 10 *stuff* classes and 6 *thing* classes. Road, sidewalk, building, fence, pole, traffic sign, fence, terrain, vegetation, and sky are the *stuff* classes. Whereas, pedestrian, car, truck, rider, bicycle, and bus are the *thing* classes. In the BDD100K-APS dataset, the number of instances of car and pedestrian classes is relatively close and are the predominant classes followed by the truck class. Bicycle and bus classes have similar instance distributions whereas instances of rider are the least with 1.1% of the total instances. Fig. 1 (b) presents the occlusion level distribution of instances of this dataset. About 54% of the instances in the dataset are not occluded or are slightly occluded. The number of instances having a higher degree of occlusion level approximately decreases with an increase in the occlusion level. In Tab. 2, the convexity-simplicity average value for the amodal segments is lower for this dataset implying BDD100K-APS is a more complex dataset due to the presence of a large number of non-rigid objects such as pedestrians. Fig. 2 depicts

Class	Car	Pedestrian	Cyclist	Two-Wheelers	Truck	Van	Other-Vehicles
Number	192624	6240	3096	2805	6561	3573	443
Ratio	89.4%	2.8%	1.4%	1.3%	3.0%	1.6%	0.2%

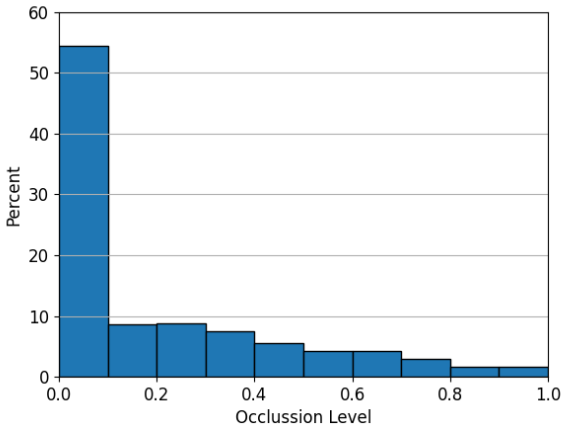
Table 2. *Thing* class distribution of KITTI-360-APS dataset.

Class	Pedestrian	Car	Truck	Rider	Bicycle	Bus
Number	19671	23775	2653	561	1110	1288
Ratio	40.1%	48.4%	5.4%	1.1%	2.3%	2.7%

Table 3. *Thing* class distribution of BDD100K-APS dataset.



(a) Occlusion level of KITTI-360-APS dataset.



(b) Occlusion level of BDD100K-APS dataset.

Figure 1. Illustration of occlusion level (defined as the fraction of region area that is occluded) in KITTI-360-APS (a) and BDD100K-APS (b) datasets.

examples from this dataset.

2. Baseline Architectures

We introduce a total of six baselines for our proposed amodal panoptic segmentation task. We create the baselines by building upon the EfficientPS [9] model which is a state-of-the-art top-down panoptic segmentation network. The EfficientPS architecture consists of four parts. The first part is the shared backbone which is a combination of an encoder and a feature pyramid network (FPN) variant. We employ the EfficientNet-B5 [12] model as the encoder and remove its squeeze and excitation [5] connections. We also replace the batch normalization and activation layers with synchronized Inplace Activated Batch Normalization (iABN sync) [1] and Leaky ReLU activations respectively. The backbone uses the 2-way FPN [9] on top of the encoder to bidirectionally aggregate multi-scale features. The encoded multi-scale features from the backbone are then propagated to an instance and semantic head. The instance head is a variant of Mask R-CNN [4] where the convolution operation in the mask prediction heads is replaced by depth-wise separable convolutions. The semantic segmentation head incorporates various modules to focus on modeling of different feature representations: DPC [2] for capturing long-range contextual information, LSFE [9] for capturing characteristic features, and MC [9] for aligning mismatched correction modules. The final component of EfficientPS is an adaptive fusion module that fuses the output of instance and semantic head based on their logits.

In the baseline architectures, we keep all the components of EfficientPS intact except for the instance segmentation head which is replaced by different state-of-the-art amodal instance segmentation heads namely, Amodal EfficientPS, ORCNN [3], VQ-VAE [6], Shape Prior [13], ASN [10], and BCNet [7]. In the following, we provide a brief overview of the amodal instance segmentation heads of the baselines.

1. **Amodal-EfficientPS** is an extension of its inmodal variant and relies implicitly on the network to learn the relationship between the occluder and occludee along with modeling the appropriate class-specific structures. Fig. 3 (a) presents the amodal instance head of Amodal-EfficientPS.

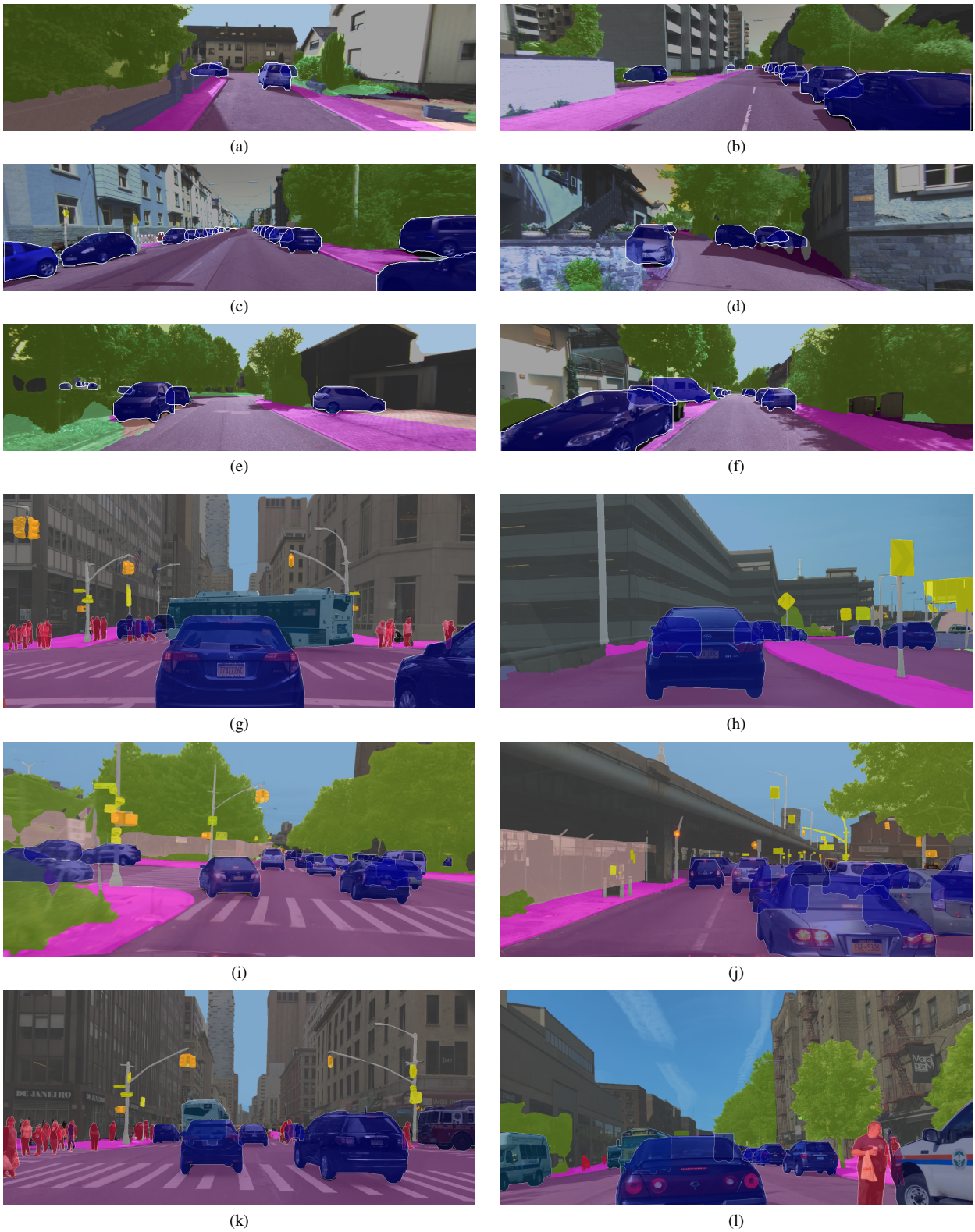


Figure 2. Visualization of amodal panoptic segmentation groundtruth from our proposed KITTI-360-APS (a-f) and BDD100K-APS (g-l) datasets. In (a) and (f) the second car on the left, (e) the far away cars on the left are heavily occluded by other car instances and vegetation, respectively. Similarly, in (h) and (l) the center cars occlude the car and the truck in front of them to a high degree, respectively. Moreover, we also observe a varying degree of occlusion from partial to mid in all of the visualization examples. The variations in occlusion of instances, cluttered urban road scenes with several *thing* class instances, and complex *stuff* classes makes both the proposed datasets extremely challenging for amodal panoptic segmentation.

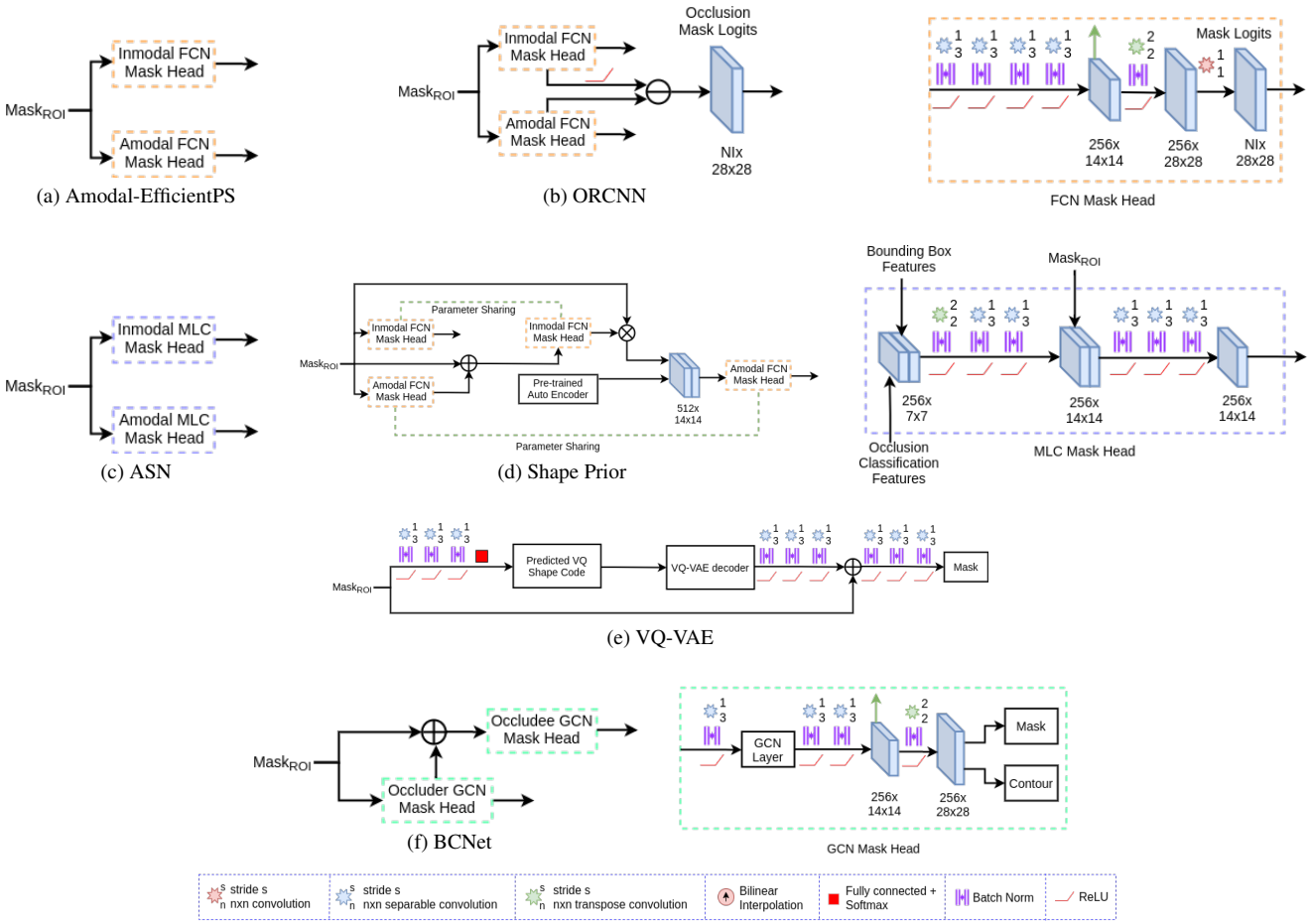


Figure 3. Topologies of various amodal instance segmentation head of the amodal panoptic segmentation baselines. Please note that the boxes enclosed in color dashes in each of the architecture corresponds to the expanded version of the same colored boxes depicted on the right.

2. **ORCNN** [3] employs an invisible mask prediction head in addition to the inmodal and amodal mask prediction heads, to explicitly learn the propagation from visible mask to amodal mask. To do so, the approach designs the invisible mask prediction by abstracting the amodal mask from the visible mask. Fig. 3 (b) shows the amodal instance head of ORCNN.
3. **ASN** [10] head emphasizes the importance of global information in addition to visible cues for amodal mask prediction. Fig. 3 (c) presents the ASN amodal instance head. It consists of an additional occlusion classification branch and uses the features from this branch through a multi-level coding (MLC) block to impart the learned global information to the individual inmodal and amodal mask prediction head. The MLC block essentially takes the concatenation of bounding box features and occlusion features from their respective classification branches, performs a series of transpose convolution-convolution operations to process the collective features, and then concatenates it with the model-specific mask features. This is followed by another series of convolution operations to generate the final modal-specific mask predictions.
4. **Shape Prior** [13] approach strongly supports the idea of using the visible region segmentation in conjunction with shape priors as the key to better amodal mask segmentation. Fig. 3 (d) depicts the amodal instance head of this approach. The aforementioned head employs two modal-specific fully convolutional network heads with parameter sharing. The first modal-specific heads give the initial mask predictions that are further used as attention for refining the final mask predictions with a feature matching loss and pre-trained shape prior autoencoder. Additionally, the approach also incorporates the shape-prior autoencoder in the non-maximum suppression step of the amodal bounding boxes [11].
5. **VQ-VAE** [6] seeks to incorporate shape prior information through discrete shape codes while using Vector Quantized Variational Autoencoder for mask segmentation. Fig. 3 (e) shows the amodal instance head of

VQ-VAE.

6. **BCNet** [7] models occluder and occludee with a bi-layer GCN layer. To be precise, the approach first predicts the occluder mask and contour segmentation and uses these occluder features in conjunction with the ROI features to segment the occludee or the target object in a class agnostic manner. Fig. 3 (f) presents the amodal instance head of this approach. In contrast, our APSNet employs FCN based class agnostic occluder mask segmentation head to coarsely model the occlusion regions of the target object as a strong prior and is further refined in a spatially independent manner with an occlusion mask segmentation head. Moreover, we use additional processing blocks with spatio-channel attention to explicitly model the underlying relationship among occluder (general location and shape of the occluded region), occludee (visible region), and the occlusion (precise shape of the occluded region) features before finally computing the amodal mask segmentation. Fig. 4 illustrates our fragmentation of the amodal bounding box of a target object.

To summarize, for a better amodal perception performance, an amodal instance head should have the ability to decipher the existence of occlusion regions and be able to reason about the shape given the visible region features. We build our APSNet on these two core ideas.

3. Inference

At inference time, to obtain the amodal panoptic segmentation output, we fuse the amodal instance segmentation and the semantic segmentation predictions. There are several fusion heuristics [8, 9, 14] that have been proposed for panoptic segmentation. We adapt the panoptic fusion proposed in [9] due to its superior performance over other fusion approaches. This heuristic allows adaptive fusion of the task-specific head outputs, which can alleviate the inherent overlap problem between the outputs of the different heads. The semantic head generates semantic logits of $|C_s| + |C_t|$ channels where C_s and C_t are the set of *stuff* and *thing* semantic classes. While the amodal instance head outputs a set of object instances consisting of a class prediction score, confidence score, amodal bounding box prediction, inmodal, and amodal mask logits. To apply the panoptic fusion, we need to compute two logits ML_A and ML_B. We begin with the computation of logit ML_A where we apply confidence thresholding to reduce the number of instances followed by the ROI sampling operation for the amodal bounding box on the two model-specific logits to increase their resolution from 28×28 to the input image resolution $H \times W$. Here, H and W are the height and width of the input image. Subsequently, we compute the inmodal bounding box from the inmodal mask derived from the inmodal mask logits. We then sort the class prediction, the modal-specific logits, and the inmodal bounding box according to the class confidence score. We then employ overlap thresholding using the inmodal mask logits to finally yield the mask logit ML_A.

We compute the second mask logit ML_B for the corresponding instances of objects from the semantic head logits by selecting the channel based on the class of the instance and zero-out the logits for that channel outside the inmodel bounding box. Lastly, we fuse the two logits ML_A and ML_B as

$$FL = (\sigma(ML_A) + \sigma(ML_B)) \odot (ML_A + ML_B), \quad (3)$$

where $\sigma(\cdot)$ is the sigmoid function and $\odot$ is the Hadamard product.

We then concatenate the *stuff* logits from the semantic head logits with FL. Subsequently, we apply softmax and the argmax operation along the channel dimension to obtain the so-called intermediate prediction (IP). In the final step, we zero out the *stuff classes* class labels and copy the semantic head prediction *stuff* labels to the zero places in IP. We obtain the amodal mask for each instance in IP by accessing the amodal mask logits channels according to the instance ID. We then compute the sigmoid of the selected amodal mask logits and threshold it at 0.5 to obtain the final amodal binary mask. Following, we set the pixels in the amodal binary mask to 2 that does not overlap with the corresponding instance ID mask in IP to represent its occluded regions. The set of this tensor along with its class prediction and instance ID is concatenated with IP to yield the final amodal panoptic prediction.

4. Amodal Instance Head

To recapitulate, our proposed amodal segmentation head aims to impart the awareness of the presence of occlusion regions with coarse localization (occluder head) and learn to perceive the occlusion shape given the visible and occluder regions. It also models the necessary interconnecting features of occlusion, occluder, and visible regions (processing block with spatio-channel attention) to be able to predict the amodal mask. Additionally, it uses the computed amodal features to further refine the inmodal mask prediction. Further, to efficiently train the occlusion mask head with dense feedback, our APSNet opts to learn spatially independent occlusion masks. Fig. 5 presents examples of the spatially dependent and independent occlusion groundtruth masks.

The amodal instance segmentation head of APSNet consists of object classification, bounding box regression, and various mask heads. The training loss for bounding box object classification head L_{cls} and the bounding box regression head L_{bbx} is the same as defined in [9]. Similarly, the visible mask head loss L^{v}_{mask}, occluder mask head loss L^{od}_{mask}, occlusion mask head loss L^{ol}_{mask}, amodal mask head

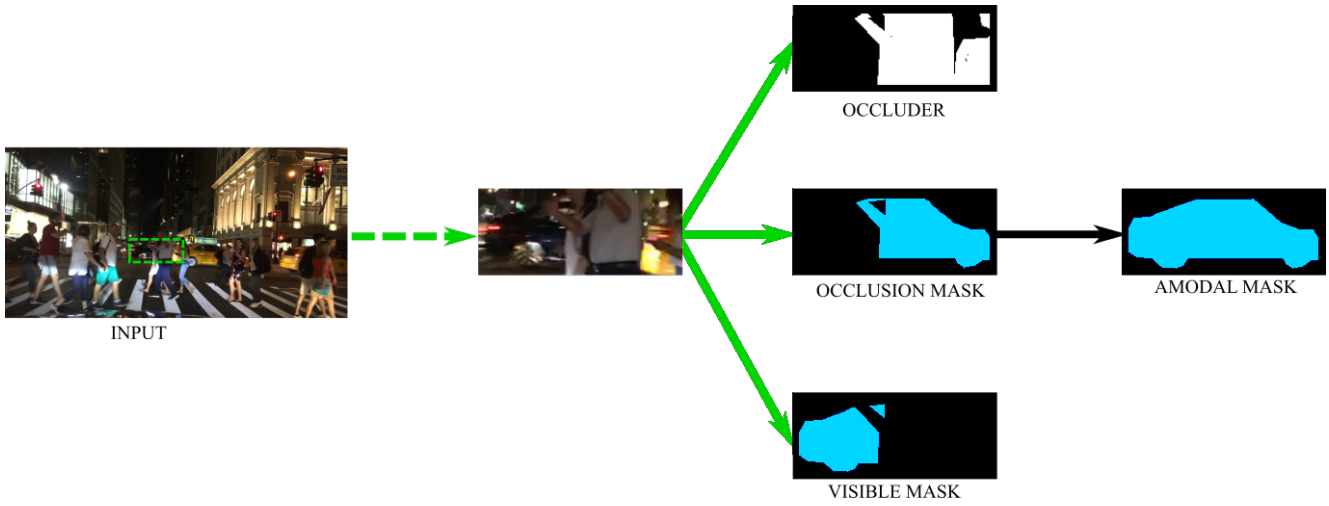


Figure 4. Illustration of our fragmentation of bounding box of the target object into class-agnostic occluder, class-wise occlusion, and visible masks. Our amodal instance head employs individual mask heads to predict each mask. The features from these mask heads are further processed with a series of convolution operations along with spatio-channel attention to predict the amodal mask of the target object.

loss L^{am}_{mask} and inmodal mask head loss L^{inm}_{mask} are akin to L_{mask} in [9] given as

$$L_{mask}(\Theta) = -\frac{1}{|K_p|} \sum_{(P,P') \in K_p} L_p(P, P'), \qquad (4)$$

where $L_p(P, P')$ is the binary cross-entropy loss, P is the ground truth binary mask, P' is the predicted binary mask and K_p is the set of positive matches.

Thus the overall loss for our proposed amodal instance segmentation head is as

$$L_{ainst} = L_{cls} + L_{bbx_{am}} + L_{bbx} + L^{v}_{mask} + L^{od}_{mask} + L^{ol}_{mask} + L^{am}_{mask} + L^{inm}_{mask}. \qquad (5)$$

Note that the gradient from the loss L_{ainst} does not flow through the RPN.

5. Extended Benchmarking Results

In this section, we discuss the benchmarking results on the KITTI-360-APS validation set in detail to understand the mutual relationship between the two proposed metrics clearly. Tab. 4 presents the quantitative results using the APQ and APC metrics and all their components. The APS baseline with the trivial implementation of amodal instance head, Amodal-EfficientPS achieves the lowest APQ and APC scores. Similarly, ORCNN that employs a derivative head for occlusion mask prediction over the trivial amodal instance head attains similar performance to Amodal-EfficientPS. However, this similar overall performance of the two networks stems from varying effectiveness of segmenting the visible and invisible regions rather than being the same. APS-EfficientPS has higher APQ^{V}_{T} and APC^{V}_{T} scores implying better visible *thing* region parsing, whereas ORCNN has higher APQ^{O}_{T} and APC^{O}_{T} values, indicating better occluded *thing* region parsing. Following, BCNet performs better than Amodal-EfficientPS and ORCNN, lagging behind VQ-VAE by 0.1% in both APQ and APC scores. However, BCNet achieves an improvement of 1.9% in APQ^{O}_{T} and 1.2% in APC^{O}_{T} scores. This difference in the proportional improvement in the two metrics where the increase in performance is higher for APQ^{O}_{T} signifies that BCNet primarily improves the segmentation of partially occluded objects with smaller occlusion regions. When paired with the improvement in APQ^{V}_{T} of 0.7% and APC^{V}_{T} of 1.2% indicates that the approach improves the segmentation of nearby larger objects that are partially occluded. We hypothesize that this is primarily due to the bilayer modeling of occluder and occludee which enables more refined target mask segmentation.

Subsequently, VQ-VAE adds an occlusion detection branch and mask refinement with shape priors to incorporate amodal reasoning capabilities. Compared to the trivial Amodal-EfficientPS, this approach achieves an improvement of 0.7% in APQ and 0.4% in APC, where the improvement in APQ_{T} and APC_{T} component of the metrics is 1.5% and 1.2% respectively. Next, the Shape Prior model refines the coarsely predicted mask with shape priors in addition but uses a combination of a pre-trained autoencoder with K-Means based codebook. It further incorporates a visible mask refinement step with amodal features. This model has an APQ score of 41.8% and an APC score of 58.2%. Its APQ^{O}_{T} and APC^{O}_{T} are higher than that of VQ-VAE by 0.6% and 0.4% respectively. This improvement suggests that incorporating shape priors with an additional codebook yields better performance. A similar trend is also observed for visible *thing* region metrics indicating that refining visible

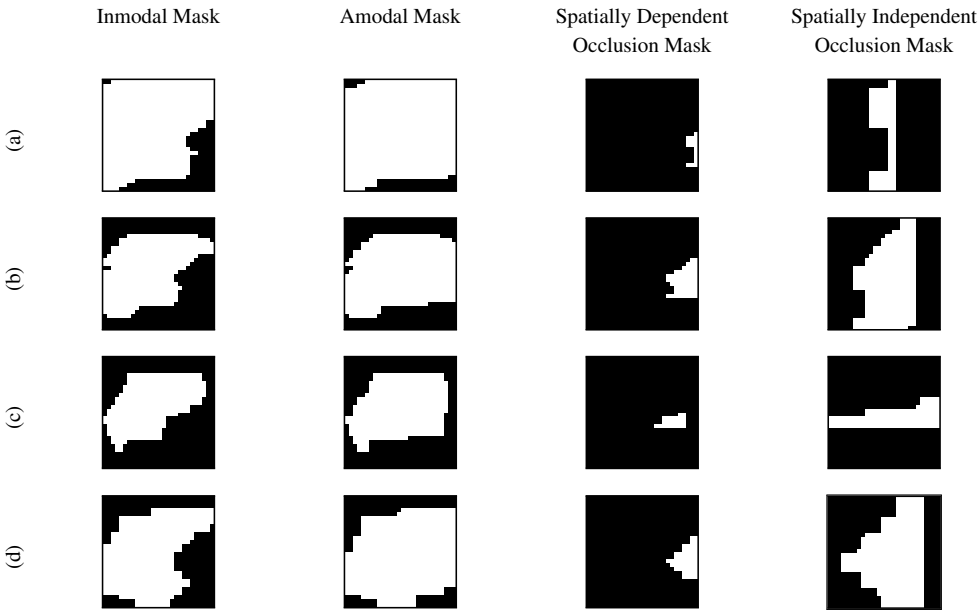


Figure 5. Illustration of spatially dependent and independent occlusion masks. The spatially dependent occlusion mask consists of few pixels compared to the inmodal mask for partial occlusion. On the other hand, the spatially independent occlusion masks that effectively capture the underlying shape of the occluded regions are denser. Thus, enabling stronger feedback during training and consequently resulting in capturing the underlying shape of the occlusion mask effectively.

	APQ	APC	APQ_S	APQ_T	APQ_T^V	APQ_T^O	APC_S	APC_T	APC_T^V	APC_T^O	AP	mIOU
Amodal-EfficientPS	41.1	57.6	46.2	33.1	41.3	12.7	58.1	56.6	58.5	22.7	29.1	44.7
ORCNN [3]	41.1	57.5	46.2	33.1	41.1	12.8	58.1	56.6	58.1	22.9	29.0	44.5
BCNet [7]	41.6	57.9	46.2	34.4	42.0	14.5	58.1	57.6	59.7	23.9	30.3	45.8
VQ-VAE [6]	41.7	58.0	46.2	34.6	42.2	14.7	58.1	57.8	59.8	23.9	30.4	45.9
Shape Prior [13]	41.8	58.2	46.2	35.0	42.5	15.3	58.1	58.2	60.3	24.3	31.0	46.3
ASN [10]	41.9	58.2	46.2	35.2	42.7	15.4	58.1	58.3	60.4	24.2	31.1	46.3
APSNet (Ours)	**42.9**	**59.0**	**46.7**	**36.9**	**43.6**	**18.3**	**58.5**	**59.9**	**61.5**	**25.8**	**33.4**	**48.0**

Table 4. Performance comparison of amodal panoptic segmentation on the KITTI-360-APS validation set. Subscripts S and T refer to *stuff* and *thing* classes respectively. Subscripts S and T refer to *stuff* and *thing* classes respectively. Superscripts V and O refer to visible and occluded regions respectively. All scores are in [%].

mask with amodal features helps improve the performance further.

Nevertheless, our proposed approach performs the best in all the metrics, namely APQ and APC, and their components. Here, the proportional improvement of APSNet can be observed in visible and occlusion components of APQ_T (0.9% and 2.9%) and APC_T (0.9% and 1.6%) compared to the best baselines ASN. This demonstrates that our approach improves the segmentation of partial-to-mid occluded objects, however, the performance is limited when it comes to heavily occluded objects, as observed in the qualitative evaluations in the manuscript. To conclude, computing both the metrics for the amodal panoptic segmentation task gives more insights into the performance of an approach, which can be extremely valuable while developing an effective solution for this problem.

References

[1] Samuel Rota Bulo, Lorenzo Porzi, and Peter Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5639–5647, 2018. 2

[2] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *Advances in Neural Information Processing Systems*, pages 8713–8724, 2018. 2

[3] Patrick Follmann, Rebecca König, Philipp Härtinger, Michael Klostermann, and Tobias Böttger. Learning to see the invisible: End-to-end trainable amodal instance segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1328–1336. IEEE, 2019. 2, 4, 7

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Gir-

shick. Mask r-cnn. pages 2961–2969, 2017. 2

[5] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[6] Won-Dong Jang, Donglai Wei, Xingxuan Zhang, Brian Leahy, Helen Yang, James Tompkin, Dalit Ben-Yosef, Daniel Needleman, and Hanspeter Pfister. Learning vector quantized shape code for amodal blastomere instance segmentation. *arXiv preprint arXiv:2012.00985*, 2020. 2, 4, 7

[7] Lei Ke, Yu-Wing Tai, and Chi-Keung Tang. Deep occlusion-aware instance segmentation with overlapping bilayers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4019–4028, June 2021. 2, 5, 7

[8] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 5

[9] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021. 2, 5, 6

[10] 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*, pages 3014–3023, 2019. 1, 2, 4, 7

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 4

[12] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 2

[13] Yuting Xiao, Yanyu Xu, Ziming Zhong, Weixin Luo, Jiawei Li, and Shenghua Gao. Amodal segmentation based on visible region segmentation and shape prior. In *AAAI Conference on Artificial Intelligence*, 2021. 2, 4, 7

[14] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 5

[15] 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