Supplementary Material for "RbA: Segmenting Unknown Regions Rejected by All"

Overview

This supplementary document contains the implementation details that are necessary to reproduce our approach (Section A), additional results using extra data (Section B), additional ablation results to justify hyper-parameter choices (Section C), additional details about analysis and ablation experiments reported in the main paper (Section D), additional qualitative results to showcase our method compared to state-of-the-art (Section E), and some challenging cases that cause failure (Section F).

A. Implementation Details

A.1. Architecture

Fig. 1 illustrates the full Mask2Former architecture along with our unknown inference procedure. The main components of the model are the backbone, the pixel decoder, and the transformer decoder. We explain the details of each next.

Backbone: We use the Swin-B variant as the backbone [9]. It can take an RGB image with any resolution higher than 32×32 as input and outputs feature maps at several resolutions to the pixel decoder. Specifically, the output feature maps are downsampled with strides 4 ($\mathbf{x}_4$), 8 ($\mathbf{x}_8$), 16 ($\mathbf{x}_{16}$), and 32 ($\mathbf{x}_{32}$) with respect to the input image.

Pixel Decoder: Following [2], the pixel decoder mainly consists of 6 layers of deformable attention (MSDeformAttn) [14]. The multi-scale feature maps with strides $\mathbf{x}_8$ $\mathbf{x}_{16}$, and $\mathbf{x}_{32}$ are processed with MSDeformAttn layers to produce $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}_3$, respectively. In [2], the three processed feature maps are passed to 9 transformer decoder layers in a round-robin fashion. However, we found that using a single layer in the transformer decoder works better for unknown detection. Therefore, we only pass the last layer $\mathbf{f}_3$ to the transformer decoder. The feature map $\mathbf{x}_4$ is processed with a 1×1 filter-size convolutional layer and then added to the processed feature map $\mathbf{f}_1$ after bilinear upsampling. Finally, the output is passed to a 3×3 convolutional layer to produce per-pixel features $\mathbf{F} \in \mathbb{R}^{C_p \times H \times W}$, where $C_p = 256$ is the embedding dimension. The computation can be summarized as follows:

$$\mathbf{F} = \text{Conv}_{3\times3}\left(\text{Conv}_{1\times1}(\mathbf{x}_4) + \text{Upsample}(\mathbf{f}_1)\right) \quad (1)$$

Transformer Decoder: Learnable object queries $\mathbf{Q} \in \mathbb{R}^{N \times C_q}$ are fed to the transformer decoder layer to be processed with the feature maps from the pixel decoder, where $N = 100$ is the number of object queries and $C_q = 256$ is the embedding dimension. A single transformer decoder layer consists of a cross-attention layer followed by self-attention and feed-forward network (FFN), each of which is followed by a LayerNorm. The cross-attention operation is performed with mask-attention, where each object query only attends to regions it predicted in the previous layer. Since we use only a single layer, each object query attends to the region it predicts directly from the input feature map before being processed in the transformer decoder. Learnable positional embeddings are added to the object queries. The transformer decoder outputs a refined set of object queries $\mathbf{Q}_r$ that predict the regions and classify them.

Region Class Prediction: The refined object queries $\mathbf{Q}_r$ are fed into a single linear layer followed by a softmax to produce the class probability of each region $\mathbf{P} \in \mathbb{R}^{N \times K}$ where K is the number of classes.

Membership Maps Prediction: The refined object queries $\mathbf{Q}_r$ are also fed into a 3-layer MLP, so that $\mathbf{Q}_r$'s dimensionality matches that of $\mathbf{F}$. Then, $\mathbf{Q}_r$ and $\mathbf{F}$ are multiplied before being fed into a sigmoid activation to produce the per-pixel membership maps $\mathbf{M} \in \mathbb{R}^{N \times H \times W}$.

A.2. Closed-Set Training

Loss Functions: Before applying any loss function, bipartite matching is used to match object queries to ground truth binary masks, where each mask contains all the pixels of a certain class. The matching cost is computed as a weighted sum of the individual losses. The classification is performed with the cross-entropy loss. A weighted combination of dice loss and binary cross-entropy is used to predict regions.

Hyper-Parameters: Following [2], the model is trained for 90K iterations using a batch size of 16. AdamW [10] optimizer is used with 0.05 for weight decay and an initial

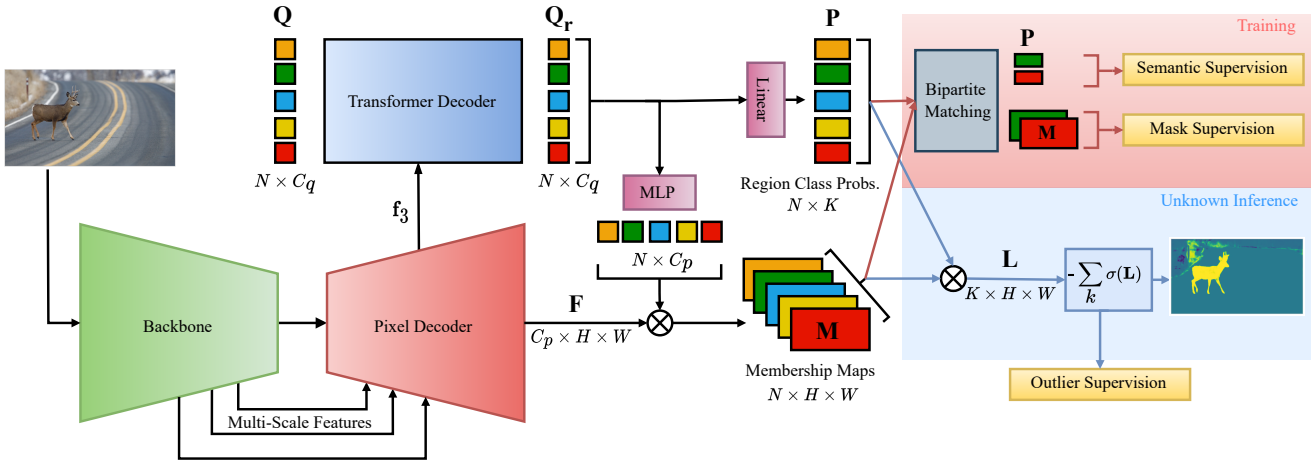


Figure 1: **Detailed architecture.** This figure provides a more detailed view of the Mask2Former [2] architecture, including our modifications and unknown inference computation. We use a single transformer decoder layer as opposed to the original implementation that uses 9 layers. Therefore, only a single scale feature $\mathbf{f}_3$ from the last layer is passed from the pixel decoder to the transformer decoder. For outlier supervision, all modules are frozen except for the MLP and the linear layers shown in pink.

learning rate of 10^{-4}, which is reduced using a polynomial scheduler. The learning rate for the backbone is multiplied by 0.1.

Data Augmentation: We use the same augmentations as in [2]. First, the short side of the input image is resized by a scale uniformly chosen between $[0.5 - 2]$. Then a random crop of size 512×1024 is applied. After that, large-scale jittering augmentation [4, 5] is applied with a random horizontal flip.

A.3. Outlier Supervision

Data Sampling: We use a slightly modified version of AnomalyMix proposed in [12] for outlier supervision. After eliminating the samples that contain Cityscapes classes [3], around 40K images remain for outlier supervision on the COCO [7] dataset. For a single fine-tuning experiment, we randomly sample 300 images and fix them throughout the entire fine-tuning phase.

Fine-tuned Components: For all the fine-tuning experiments, we only fine-tune the 3-layer MLP and linear layers shown in pink in Fig. 1. Their weights together constitute approximately 0.21% of the entire model parameters.

Hyper-Parameters: After the model is trained on the closed-set setting, we fine-tune it for 2000 iterations on Cityscapes [3] using the setting of the closed-set training; AdamW [10] optimizer with 0.05 weight decay and 10^{-4} initial learning with polynomial scheduling. For every Cityscapes image used in fine-tuning, an object from the 300 COCO samples is uniformly chosen and pasted on the Cityscapes image with probability $p_{out} = 0.1$, which is independent for each image. The RbA score for outlier pixels is optimized with the squared hinge loss $\mathcal{L}_{RbA}$ using $\alpha = 5$.

Model	Anomaly Track		Obstacle Track	
	AP	FPR	AP	FPR
RbA (Swin-B)	94.46	4.60	93.68	0.15
RbA (Swin-L)	93.78	4.59	95.12	0.08

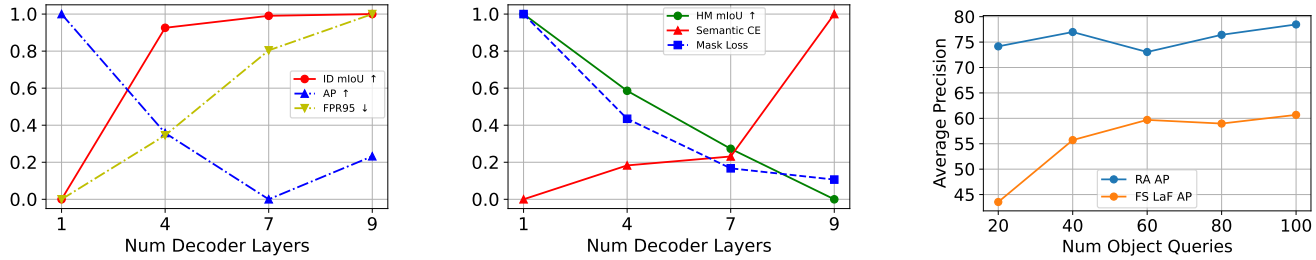
Table 1: **Results on SMIYC using additional data from Mapillary dataset**

B. Training with Extra Inlier Data

We show additional results on the SMIYC benchmark by using extra training inlier training data from the Mapillary Dataset [11]. We train two different backbones (Swin-L & Swin-B) on both cityscapes and Mapillary after mapping the Mapillary semantic classes to match the cityscapes taxonomy. As shown in Table 1, additional inlier training results in noticeable improvements in performance. This highlights the positive correlation between the ability to segment known classes and RbA's performance.

C. Additional Ablation Study

Number of Transformer Decoder Layers: As shown in [2], more transformer decoder layers improve the inlier performance, i.e. mIoU on Cityscapes. However, we found that using fewer decoder layers results in better performance in terms of outliers. Fig. 2a highlights the decrease in performance in terms of the AP and FPR@95 on the Road



(a) Number of Decoder Layers (b) Behavior of Different Losses (c) Number of Object Queries

Figure 2: **Ablation study on architectural choices.** With more decoder layers, in-distribution performance on Cityscapes improves but the outlier performance drops in terms of AP and FPR@95 on Road Anomaly a. Good performance is mainly due to better mask prediction at the cost of a higher semantics loss b. The drop in mIoU with hard masking (**HM mIoU**) is another indicator of semantic information loss in the object queries with more decoder layers. The performance improves as the number of object queries increases c.

Num Iter	AP ↑	FPR@95 ↓
1000	83.20 ± 0.11	8.69 ± 0.04
2000	85.49 ± 0.08	**7.25** ± 0.04
3000	**85.72** ± 0.12	7.82 ± 0.11
4000	84.89 ± 0.07	7.45 ± 0.05
5000	84.66 ± 0.06	8.14 ± 0.03

(a) Number of Iterations

p_{out}	AP ↑	FPR@95 ↓	mIoU ↑
0.05	84.34 ± 0.15	9.12 ± 0.10	**82.28** ± 0.02
0.1	85.48 ± 0.12	**7.24** ± 0.06	82.15 ± 0.09
0.2	**85.74** ± 0.11	7.90 ± 0.11	81.54 ± 0.02
0.4	84.97 ± 0.11	9.19 ± 0.05	81.27 ± 0.06

(b) Outlier Selection Probability

α	AP ↑	FPR@95 ↓
-0.1	84.01 ± 0.18	9.25 ± 0.06
-0.01	84.41 ± 0.14	9.06 ± 0.11
0.0	84.30 ± 0.06	9.10 ± 0.07
2	85.30 ± 0.07	7.80 ± 0.06
5	85.24 ± 0.08	**6.95** ± 0.08
10	**85.61** ± 0.10	7.26 ± 0.07

(c) Outlier Threshold

Table 2: **Ablation study on outlier supervision.** Finetuning with RbA loss for 2000-3000 iterations achieves the best performance and the performance deteriorates after 3000 iterations as shown in a. A higher probability of exposure to outlier data results in a consistent decline in the closed-set performance. A probability of 0.1 achieves the best balance between outlier and inlier performance b. Finally, we ablate the RbA loss parameter α in c and find that the best results are achieved with $\alpha > 0$.

Module	Params (%)	mIoU	Road Anomaly		FS LaF	
			AP ↑	FPR ↓	AP ↑	FPR ↓
Full Model	100	80.81	76.00	9.50	**73.88**	6.02
Transformer Dec.	1.93	80.24	85.08	10.18	72.6	**5.51**
Pixel Dec.	4.66	81.59	75.83	10.51	69.8	6.77
MLP+Linear	**0.21**	**82.20**	**85.42**	**6.92**	70.81	6.30

Table 3: **Ablation study on fine-tuning different modules.** We show the effect of fine-tuning different components of the model on the Road Anomaly and Fishyscapes LaF validation sets. Fine-tuning MLP+Linear maintains the best performance in unknown detection without sacrificing the closed-set performance.

Anomaly dataset as the number of decoder layers increases. We investigate this behavior by isolating the sources of error with respect to the number of decoder layers. Fig. 2b shows semantic and mask losses of the Mask2Former [2] averaged over the validation samples on Cityscapes. With more decoder layers, we observe that the semantic cross-entropy loss increases while the mask-related BCE and dice losses decrease. This shows that the increase in inlier mIoU with more decoder layers can be attributed to increased performance in detecting masks at the cost of a higher semantic error. By using fewer decoder layers, we regulate the semantic confusion, which helps to better align the logit scores, resulting in better outlier performance.

Fig. 2b also shows the mIoU evaluated by applying hard masking on the specialized object queries. Specialized object queries perform worse with more decoder layers. The information loss in the object queries as well as the increase in the semantic loss show the importance of semantics for outlier segmentation compared to precise masks.

Number of Object Queries: The original Mask2Former [2] uses 100 object queries. We train different models by varying the number of object queries to observe its effect on detecting outliers. We focus on the AP values on both Road Anomaly [8] with large objects and on Fishyscapes LaF [1] with small objects. Fig. 2c shows that more object queries result in better AP on both datasets. Even if some object queries specialize in predicting a particular class, the other object queries still play a role, especially for rare classes, as

shown in Fig.3 in the main paper.

Outlier Data Exposure: We perform an experiment to show the effect of the number of iterations required for fine-tuning with our RbA loss in Table 2a. We report the AP and FPR metrics on the Road Anomaly dataset, averaged over 5 different runs to eliminate the effect of randomness. We can see that the best results can be achieved after around 2000 and 3000 iterations and then begin to degrade. We also evaluate the effect of the amount of outlier data exposed during training, which is controlled by the parameter p_{out}. We experiment with different values for p_{out} and report the outlier performance on Road Anomaly and closed-set performance on Cityscapes averaged over 5 different runs in Table 2b. We can see that more exposure to outlier data negatively affects the closed-set performance. Consequently, even the outlier segmentation performance starts to degrade for $p_{out} > 0.2$. We choose the $p_{out} = 0.1$ because it strikes a reasonable balance between outlier and closed-set performance.

RbA Loss Parameter: In Table 2c, we report the performance of our loss function $\mathcal{L}_{\text{RbA}}$ using different values of α, averaged over 5 different runs. We find that positive values of α work similarly well and set α to 5 in our experiments.

Fine-tuned Component: In Table 3, we analyze the effect of fine-tuning different parts of the model on validation sets of Road Anomaly [8] and Fishyscapes Lost and Found [1] using RbA loss. The alternative components we experimented with are the full model, only the transformer decoder (blue + pink in Fig. 1), only the pixel decoder (red in Fig. 1), and MLP+Linear layers (pink in Fig. 1). On the Road Anomaly dataset, fine-tuning MLP+Linear achieves the best performance in terms of AP and FPR. On Fishyscapes LaF, the best AP is achieved by fine-tuning the entire model, while the best FPR is obtained by fine-tuning only the transformer decoder. Both options cost a decrease in performance on Road Anomaly and negatively affect the closed-set performance. Fine-tuning MLP+Linear achieves the best balance between outlier detection and closed-set performance and is the least costly option in terms of the number of parameters finetuned.

D. Details of Experiments

In this section, we provide further illustrations and detailed settings of our analysis and ablation experiments in the main paper. We first perform an experiment to verify the specialization of object queries. We then provide a detailed formulation of our ablations on the loss functions and other methods using Mask2Former including the hyper-parameters that we use to obtain the results presented in the main paper.

D.1. Specialization of Object Queries

Our method is based on our finding that the object queries in mask classification enjoy a degree of independence from one another and that each object query in a subset specializes in segmenting a specific class from the closed set. Due to bipartite matching being applied between queries and ground truth class masks during training, this behavior can be anticipated. Here, we empirically verify it using a different dataset than the one used in training (BDD100K [13]). For each object query, we count how many times it predicts a certain class with high confidence. Fig. 3 shows the heatmap of counts for the Mask2Former model with 100 object queries. For each of the closed-set classes, there is a single object query dominantly predicting it.

In the main paper, we test the independence of the specialized queries by applying hard masking and soft masking and evaluating per class IoU on Cityscapes. Fig. 4 shows an illustration of hard and soft masking applied.

D.2. Other Loss Functions

In the main paper, we perform an ablation study with different loss functions in comparison to squared hinge loss. In this section, we provide the formulation and the parameter setting of each loss function. In the following, Ω_{out} denotes the set of outlier pixels, and Ω_{in}, the set of inlier pixels on an image.

Mean-Squared Error (MSE): With MSE loss, we optimize the RbA to be closer to $\alpha = 5$ for outlier pixels:

$$\mathcal{L}_{\text{MSE}} = \sum_{\mathbf{x} \in \Omega_{out}} (\text{RbA}(\mathbf{x}) - \alpha)^2 \tag{2}$$

L1: Similar to MSE, we optimize the RbA with L1 loss by setting α to 5:

$$\mathcal{L}_{L1} = \sum_{\mathbf{x} \in \Omega_{out}} |\text{RbA}(\mathbf{x}) - \alpha| \tag{3}$$

Binary Cross Entropy (BCE): We formulate the scoring of outliers as a per-pixel binary classification problem, where outliers correspond to the positive class. We use the RbA score as the logit score for the positive class. Assuming that y corresponds to the binary label (outlier vs. inlier) of a pixel $\mathbf{x}$, we optimize the BCE loss as follows:

$$\mathcal{L}_{\text{BCE}} = \sum_{\mathbf{x} \in \Omega_{out}} y \cdot \log\left(\text{RbA}(\mathbf{x})\right) + (1-y) \cdot \log\left(1 - \text{RbA}(\mathbf{x})\right) \tag{4}$$

KL Divergence: We use the KL Divergence to minimize the distance between the predicted class distribution to a fixed distribution. For inlier pixels, we minimize the distance to the Dirac delta function where the correct class has a probability of 1.0. For outlier pixels, we minimize the distance to a uniform distribution where the entropy is maximum. Even though this loss function does not optimize our proposed score function RbA, it helps us estimate the

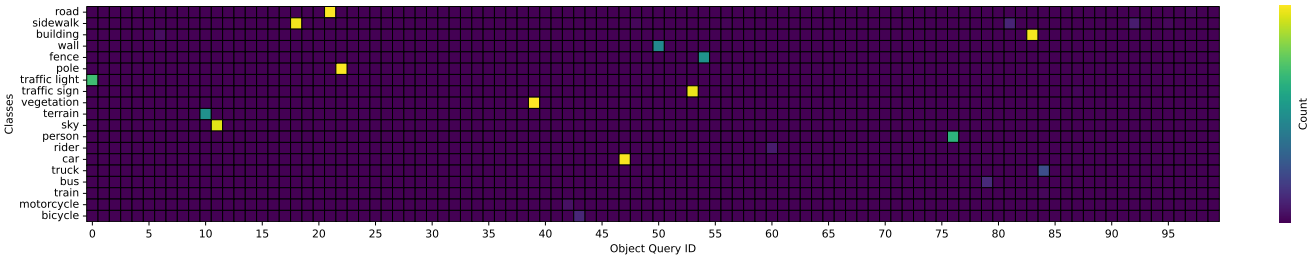


Figure 3: **Specialization of Object Queries.** Certain object queries specialize in predicting a specific class. For each query, we show how many times it predicts a region to belong to a class with high confidence. The sparsity in the plot clearly shows the specialization of queries.

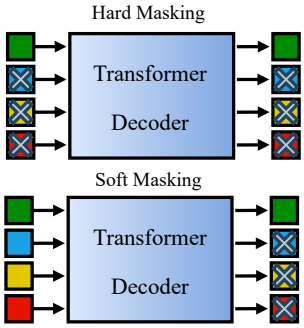


Figure 4: **Illustration of hard vs. soft masking of object queries.** In hard masking, when predicting class k, all but the object queries specialized to predict class k are masked before the transformer decoder so that the specialized query can only interact with the image features. In soft masking, the queries are allowed to interact within the transformer decoder but are dropped after, and only the specialized query is used to predict class k.

contribution of RbA by pushing the predicted class probabilities toward the ideal distributions for outliers and inliers. Let $\mathbf{L}_p(\mathbf{x})$ be the class probability distribution of a pixel $\mathbf{x}$ with ground truth label y. Let $\mathbf{P}_{in}(y)$ be a fixed probability distribution where class y has probability 1.0. Let $\mathcal{U}$ be a fixed uniform probability distribution. Formally, the loss function is defined as follows:

$$\begin{aligned} \mathcal{L}_{in} &= \sum_{\mathbf{x}\in\Omega_{in}} D_{KL}(\mathbf{L}_p(\mathbf{x}) \parallel \mathbf{P}_{in}(y)) \\ \mathcal{L}_{out} &= \sum_{\mathbf{x}\in\Omega_{out}} D_{KL}(\mathbf{L}_p(\mathbf{x}) \parallel \mathcal{U}) \\ \mathcal{L}_{KL} &= \frac{1}{2}\left(\mathcal{L}_{in} + \mathcal{L}_{out}\right) \end{aligned} \tag{5}$$

D.3. Other Methods with Mask2Former

To disentangle the contribution of our scoring function RbA from the architecture, in the main paper, we present the results of the state-of-the-art methods PEBAL [12] and DenseHybrid [6] using the Mask2Former architecture. Here, we provide the details of this ablation experiment for each method. For a fair comparison, we train both methods by fine-tuning the same components as the RbA, that is the MLP+Linear layers as shown in Fig. 1, and also using the same outlier data supervision method as described in the main paper.

PEBAL: The optimized objective as per the official implementation [12] consists of three components. First, there is the pixel-wise anomaly abstention loss (PAL) with the abstention term and penalty defined as follows:

$$\mathcal{L}_{PAL} = -\sum_{\mathbf{x}\in\Omega} \log\left(\mathbf{L}_y(\mathbf{x}) + \frac{\mathbf{L}_{K+1}(\mathbf{x})}{a(\mathbf{x})}\right)$$

where Ω is the set of all pixels on a given image, $y \in 1,\dots,K+1$ is the ground truth class of pixel $\mathbf{x} \in \Omega$, and $K+1$ is the class for the outliers. In Mask2Former, class $K+1$ is assumed to be the no object class, therefore we avoid dropping it from the region class probabilities term $\mathbf{P}$ (see Fig. 1) while fine-tuning with the PEBAL objective. The abstention penalty term $a(\mathbf{x})$ is defined as follows:

$$\begin{aligned} a(\mathbf{x}) &= (-E(\mathbf{x}))^2 \\ E(\mathbf{x}) &= -\log\sum_{k=1}^{K}\exp(\mathbf{L}_k(\mathbf{x})) \end{aligned} \tag{6}$$

where $E(\mathbf{x})$ is the free energy function. When the penalty term is high, the prediction is discouraged from abstaining and vice versa.

The second component of the loss optimizes the energy terms such that it is maximized for outlier pixels and minimized for inlier pixels as follows:

$$\begin{aligned} \mathcal{L}_{energy} = &\sum_{\mathbf{x}\in\Omega_{in}} \max(0, E(\mathbf{x}) - m_{in})^2 + \\ &\sum_{\mathbf{x}\in\Omega_{out}} \max(0, m_{out} - E(\mathbf{x}))^2 \end{aligned} \tag{7}$$

where m_{in} and m_{out} are hyper-parameters to be set. In our experiments, we experimentally use $m_{out} = -2.5$ and $m_{in} = -3.5$.

The last component is a regularization term for the smoothness and sparsity of the predicted energy map:

$$\mathcal{L}_{reg} = \sum_{\mathbf{x}\in\Omega} \beta_1 |E(\mathbf{x}) - E(\mathcal{N}(\mathbf{x}))| + \beta_2 |E(\mathbf{x})| \quad (8)$$

where $\mathcal{N}(\mathbf{x})$ is the set of vertical and horizontal neighboring pixels of $\mathbf{x}$, and β_1 and β_2 are hyper-parameters. We use $\beta_1 = 3 \times 10^{-7}$ and $\beta_2 = 5 \times 10^{-5}$.

The full objective is defined as the weighted sum of the three loss functions:

$$\mathcal{L}_{PEBAL} = \mathcal{L}_{PAL} + \beta \mathcal{L}_{energy} + \mathcal{L}_{reg} \quad (9)$$

where we set $\beta = 0.1$. Starting from the same checkpoint that we use fine-tuning RbA, we optimize the model with $\mathcal{L}_{PEBAL}$ for 5K iterations. We set other hyper-parameters to be the same as the ones that we use for RbA.

DenseHybrid: Following the official implementation of DenseHybrid [6], we added an additional outlier prediction head $\mathbf{D}(\mathbf{x}) \in \mathbb{R}^{2\times H\times W}$ to the Mask2Former model which is defined as follows:

$$\mathbf{D}(\mathbf{x}) = \mathrm{Conv}_{3\times 3}\left(\mathrm{ReLU}(\mathrm{BatchNorm}(\mathbf{x}))\right) \quad (10)$$

The outlier prediction head takes the output feature map of the highest resolution from the pixel decoder and predicts a binary output for every pixel denoting the probability of being an outlier. The objective for DenseHybrid is defined as follows:

$$\mathcal{L}_{DH} = \mathrm{CE}(\mathbf{L}(\mathbf{x}_{in}), \mathbf{Y}_{in}) + \beta_1 \mathrm{CE}(D(\mathbf{x}), \mathbf{Y}_{out}) + \beta_2 \mathcal{L}_o \quad (11)$$

where CE is short for the cross-entropy loss, $\mathbf{x}_{in} \in \Omega_{in}$ denotes the set of inlier pixels, $\mathbf{Y}_{in}$ denotes the ground truth for the closed-set and $\mathbf{Y}_{out}$ denotes the binary map where the outlier pixels are set to one. We experimentally set the hyper-parameters $\beta_1 = 0.3$ and $\beta_2 = 0.03$. $\mathcal{L}_o$ is defined as follows:

$$\mathcal{L}_o = \frac{1}{|\Omega_{out}|} \sum_{\mathbf{x}\in\Omega_{out}} \log \sum_{k=1}^{K} \exp\left(\mathbf{L}_k(\mathbf{x})\right) + \mathrm{sg}[\mathrm{mean}(\mathbf{L}(\mathbf{x}))] \quad (12)$$

where sg is short for the stop gradient operation, and mean denotes the mean of all the elements of the input tensor.

E. Additional Qualitative Results

In Fig. 5 and Fig. 6, we show additional qualitative results of RbA compared to the state-of-the-art methods PEBAL [12] and DenseHybrid [6]. For other methods, we show both the outputs of the models reported in their respective repositories and our implementations of the methods using Mask2Former. The proposed scoring function RbA reduces the false positives on the boundaries of inliers and ambiguous background regions compared to the baselines. These improvements can be observed more prominently on the obstacle track (Fig. 6) under adverse weather and lighting conditions. Moreover, compared to the baselines, RbA results in fewer false positives as a result of reducing confusion with inlier classes.

F. Failure Cases

We analyze some failure cases of our method in this section. A common reason for the failure cases is the high similarity to the inlier classes.

Tractors and Boats: As shown in Fig. 7, RbA fails to detect tractors and boats as outliers due to their similarity to inlier vehicle instances. Although the objects are partially identified, the model cannot decisively predict the whole object regions as outliers. The existing methods either segment the boats and tractors at the cost of more false positives, as in the case of PEBAL, or also suffer from a lack of smoothness, as in the case of DenseHybrid.

Far away Animals: As shown in Fig. 8, animals that are situated relatively far from the camera are confused as the inlier pedestrian class. This can be attributed to the dominance of pedestrian class in the training data as well as the similarity of legged animals to a pedestrian in appearance.

Toy Cars: Fig. 9 shows that the model fails to detect a toy car on the road and predicts it confidently as the inlier car class. While the class assignment can be considered semantically correct, it is still a hazard in a real driving scenario. Note that a small car can either be a toy car or a real car that is far away. Therefore, distinguishing real cars from toy cars might require additional information such as depth or scale.

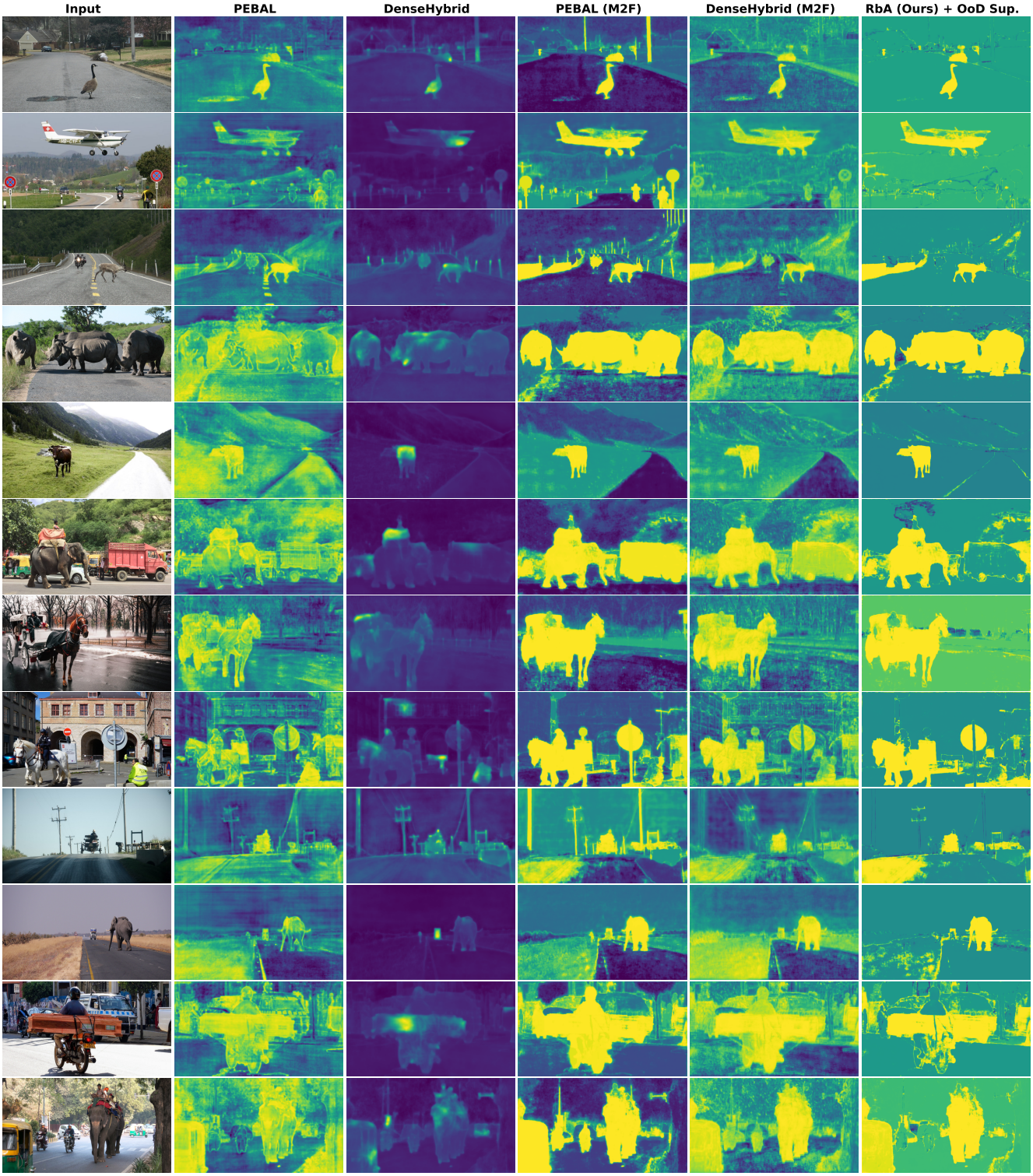


Figure 5: **Qualitative Results on SMIYC Anomaly Track.** On the anomaly track of the SMIYC benchmark, we compare RbA with outlier (OoD) supervision to the state-of-the-art methods PEBAL [12] and DenseHybrid [6] using the models that were shared in their respective repositories, as well as the versions we trained using Mask2Former (M2F). RbA better distinguishes outliers from inliers and produces more smooth outlier maps with fewer false positives.

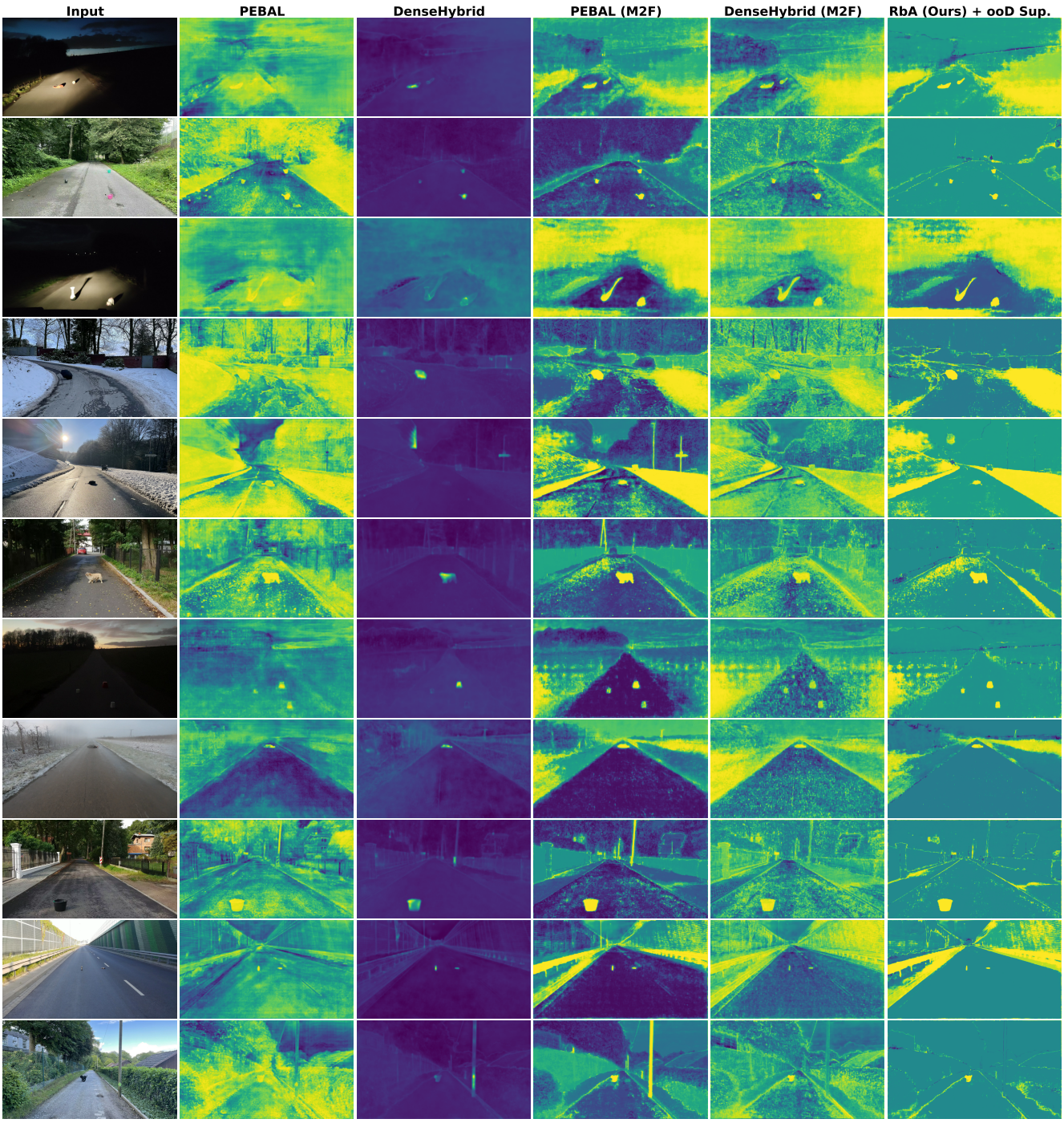


Figure 6: **Qualitative Results on SMIYC Obstacle Track.** We compare RbA with outlier supervision to the state-of-the-art methods PEBAL [12] and DenseHybrid [6]. Under adverse weather and difficult lighting conditions, RbA can detect anomalies consistently better compared to DenseHybrid and reduce false positives more compared to PEBAL.

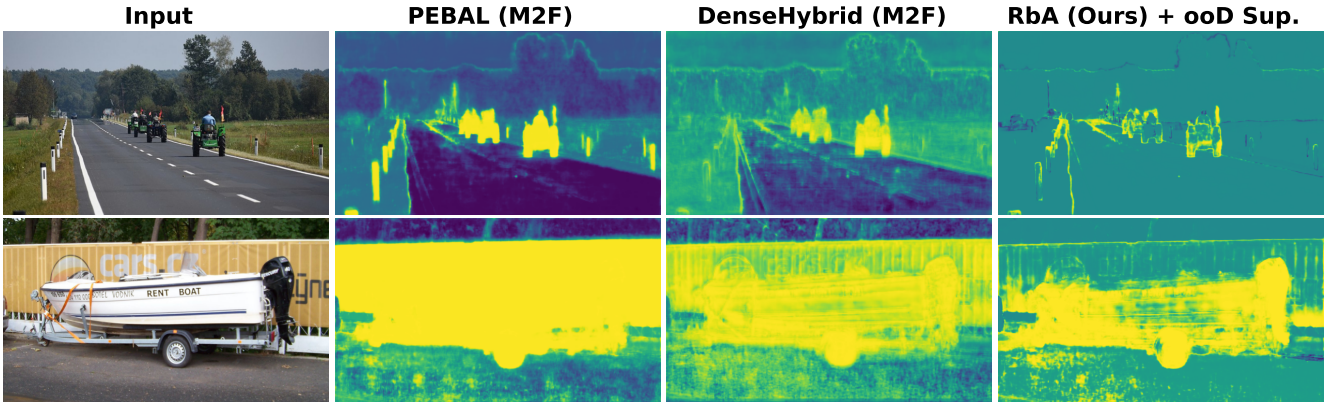


Figure 7: **Failure Cases: Tractors and Boats.** Due to their high similarity to inlier car and truck classes, unknown objects like tractors or boats are sometimes predicted as inliers.

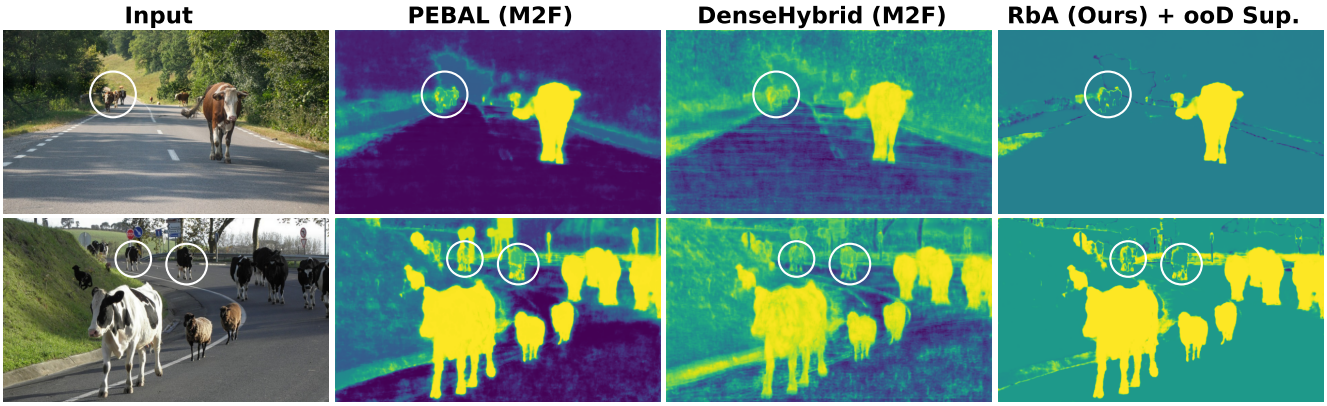


Figure 8: **Failure Cases: Animals Confused As Pedestrians.** As the pedestrian is one of the most frequent classes on Cityscapes, the model sometimes predicts animals that appear at a distance as pedestrians (highlighted in circles) on images from SMIYC Anomaly Track.

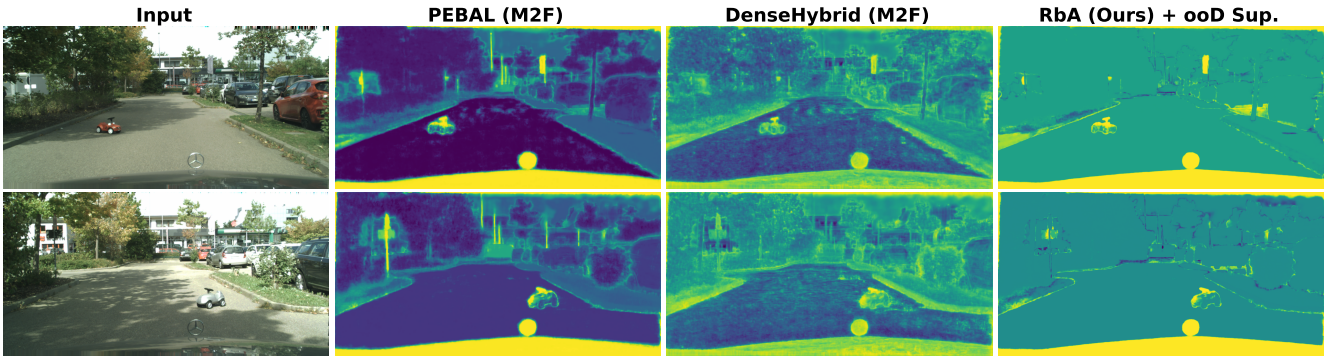


Figure 9: **Failure Cases: Toy Cars Predicted As Real Cars.** One confusing anomaly case for our model is small toy cars placed in front of the vehicle. Even though they can be semantically considered as cars, they are considered obstacles in a real driving scenario.

References

[1] Hermann Blum, Paul-Edouard Sarlin, Juan I. Nieto, Roland Y. Siegwart, and César Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *IJCV*, 129:3119–3135, 2021. 3, 4

[2] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3

[3] 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 *CVPR*, 2016. 2

[4] Xianzhi Du, Barret Zoph, Wei-Chih Hung, and Tsung-Yi Lin. Simple training strategies and model scaling for object detection. *arXiv.org*, 2107.00057, 2021. 2

[5] Golnaz Ghiasi, Yin Cui, A. Srinivas, Rui Qian, Tsung-Yi Lin, Ekin Dogus Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *CVPR*, 2021. 2

[6] Matej Grcić, Petra Bevandić, and Siniša Šegvić. Densehybrid: Hybrid anomaly detection for dense open-set recognition. In *ECCV*, 2022. 5, 6, 7, 8

[7] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 2

[8] Krzysztof Lis, Krishna Kanth Nakka, Pascal V. Fua, and Mathieu Salzmann. Detecting the unexpected via image resynthesis. In *ICCV*, 2019. 3, 4

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 1, 2

[11] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017. 2

[12] Yu Tian, Yuyuan Liu, Guansong Pang, Fengbei Liu, Yuanhong Chen, and G. Carneiro. Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. In *ECCV*, 2022. 2, 5, 6, 7, 8

[13] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *CVPR*, 2020. 4

[14] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1