

---

# UNCOVER: Unknown Class Object Detection for Autonomous Vehicles in Real-time

---

Lars Schmarje <sup>†, ‡</sup>, Kaspar Sakmann <sup>†</sup>, Reinhard Koch <sup>‡</sup>, Dan Zhang <sup>†</sup>

<sup>†</sup>Bosch Center for Artificial Intelligence

<sup>‡</sup> Kiel University

science@schmarje-sh.de

## Abstract

Autonomous driving (AD) operates in open-world scenarios, where encountering unknown objects is inevitable. However, standard object detectors trained on a limited number of base classes tend to ignore any unknown objects, posing potential risks on the road. To address this, it is important to learn a generic rather than a class specific objectness from objects seen during training. We therefore introduce an occupancy prediction together with bounding box regression. It learns to score the objectness by calculating the ratio of the predicted area occupied by actual objects. To enhance its generalizability, we increase the object diversity by exploiting data from other domains via Mosaic and Mixup augmentation. The objects outside the AD training classes are classified as a newly added out-of-distribution (OOD) class. Our solution **UNCOVER**, for UNknown Class Object detection for autonomous VEHicles in Real-time, excels at achieving both real-time detection and high recall of unknown objects on challenging AD benchmarks. To further attain very low false positive rates, particularly for close objects, we introduce a post-hoc filtering step that utilizes geometric cues extracted from the depth map, typically available within the AD system.

## 1 Introduction

Object detection is a core task in the perception stack of autonomous driving (AD) systems, in which a model is trained to recognize objects from a pre-specified list of classes typical for AD, e.g., vulnerable road users, vehicles, traffic signs and traffic lights. However, achieving a high detection performance on these training classes alone is not sufficient; whenever an object hinders safe driving, a positive detection is required, even if it is from an unknown class, i.e., an out-of-distribution (OOD) object. It is particularly challenging to address such unknown object detection for AD. Since autonomous vehicles often need to make swift decisions with limited on-device computation resources, the extra capability of unknown object detection must be acquired with a small complexity and inference latency increase [13]. Moreover, improving the recall of unknown objects should not compromise the performance on the known classes, and even more importantly should not yield many false positive detections, as they can be equally dangerous in AD scenarios, e.g. for emergency brake systems.

Current research on recognizing unknown objects in AD scenes focuses mainly on semantic segmentation [34, 28, 12, 30], with well established benchmarks like FishyScapes [1] and SegmentMeIfyouCan [5]. Compared to semantic segmentation, object detection typically requires less inference-time complexity and can localize each instance individually. However, there is less focus on equipping real-time object detectors for AD with awareness of unknown objects. Beyond the AD domain, open-world object detection leverages unannotated objects in the training set, but real-world unknown objects might be completely novel. Another line of work identified that some objects outside the training classes may still be localized by the object detector, but misclassifying them as known

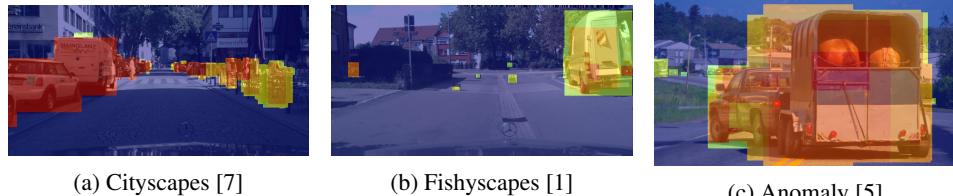


Figure 1: Visualization of occupancy – Here, UNCOVER was trained on Cityscapes [7]. Color code from blue, yellow, to red means low, medium and high occupancy. (a) UNCOVER predicts the highest occupancy on the known object classes from Cityscapes [7], e.g., vehicle, person. (b) As OOD objects, the boxes on the ground from FishyScapes [1] also have relatively high occupancy scores, compared to the background in blue. (c) It also responds to occluded objects, i.e., both vehicles and OOD object (trailer) from Anomaly [5].

classes deteriorates detection performance [8]. Our work focuses on improving the recall of unknown objects, complementing these efforts.

Aiming at real-time and low-complexity solutions, we hypothesize that the key to detecting unknown objects lies in learning a generic sense of objectness from the available training objects. In prior work on generic object detection [21, 18], the Intersection-over-Union (IoU) score, initially proposed for quantifying bounding box quality, was leveraged as an objectness measure for detecting unknowns. In [23], it was added to the one stage object detector FCOS [35] for unknown-aware multi-class object detection. In this work, we propose a new objectness measure. Unlike the IoU score, the proposed occupancy score focuses less on bounding box quality, which typically needs supervision to be high. Instead, it focuses on evaluating whether the predicted area contains one or multiple objects, relaxing the localization quality constraint. We can observe from Figure 1 that the occupancy score responds to objects from both the known and unknown classes, while it remains silent for *stuff* classes (road, sky). As exposure to diverse objects enhances the acquisition of a more generic understanding of objectness, we enrich AD training with data from MS COCO [26] and LVIS [15], using common augmentation techniques like Mosaic [2] and Mixup [41]. The former composes multiple images from different datasets into one image, providing exposure to OOD objects, while the latter interpolates the composed image with another AD image, helping to mitigate the domain gap between data domains. For objects outside the AD training cases, we introduce an extra class, i.e., OOD class, in the classification head of the base object detector.

Overall, our architecture modification consists only of an extra class in the classification head and an occupancy prediction in the regression head, resulting in a small complexity increase. We call the resulting model UNCOVER. We further propose a post-hoc and optional filtering. Depth information, often available in AD systems, encodes geometric cues of objects, which complements the RGB-based appearance cues. Applying classic computer vision algorithms for depth change detection [33, 11] helps to remove ghost object detection, such as drawings or shadows, as they don't have a geometric shape. This depth-based filtering is an interpretable algorithm helpful in reducing near-range false positive detections, thereby improving the safety of AD systems. The architecture changes and the post-hoc filtering are modular by design to allow easier adoption but we show that the best results can be achieved with UNCOVER when combining them.

We extend our evaluation to include anomaly segmentation benchmarks [1, 5] by converting the respective mask annotations into bounding box formats for analysis and assess false positives within specific regions on Cityscapes [7] and BDD100k [38].

In summary, this paper makes the following contributions for real-time AD:

- We propose a novel solution UNCOVER that enables unknown-awareness in object detection in real-time. It achieves strong recall of unknown objects on Cityscapes [7], BDD100k [38], Fishyscapes [1], and SegmentMeIfyouCan [5], with up to 25% improved recall over YoloWorld and a very limited complexity increase.
  - We further propose a depth-based post-hoc strategy to reduce false positive detections. This strategy is based on interpretable classical computer vision techniques and can be applied to any object detector. On average, we could reduce false the positive rate by 18.4% while boosting recall by 4.1%.

- We extend anomaly segmentation benchmarks for evaluation. Specifically, we make segmentation masks usable for object detection and define a false positive metric based on region of interests, such as drivable area based on the road mask.

## 2 Related Work

**Anomaly segmentation in AD** To improve AD safety, one line of work is to perform anomaly segmentation, generating a binary mask for all unknown objects in the scene. Similar to semantic segmentation, the generated mask does not localize every unknown instance separately. As high-quality semantic segmentation typically relies on a heavy pixel decoder and high-resolution feature maps, anomaly segmentation building on top of standard semantic segmentation architectures, e.g., [34, 28, 30], is typically less computationally efficient than object detection networks like YOLOs [3]. A more recent line of work on real-time panoptic segmentation [37, 39] aim at including some segmentation like lane detection into the object detection while preserving the real-time performance. However, they still lack anomaly segmentation capability. Our method is based on object detection. Inspired by anomaly segmentation, we introduce an occupancy score to indicate if the predicted bounding box contains some parts of objects. As shown in Figure 1, the occupancy map can separate foreground things and background stuffs without the necessity of a pixel-wise segmentation and not being restricted to OOD detection on the road only.

**OOD detection in object detection** Object detection targets two tasks, i.e., classification and localization. Unknown objects may still be localized. Without unknown awareness, they will be mapped to the training classes, degrading the average precision of the known classes [8]. Therefore, one line of work focuses on avoiding such misclassification, leveraging image-level OOD detection techniques to better separate novel objects from the training classes in the classification head of object detectors [10, 9]. We focus on a different challenge which is to improve the recall of unknown objects, as they may not even be localized in the first place. Misclassifying an unknown risky object as one of traffic participant classes may still result in similar planning and decision making. Overlooking them can be more critical.

**Open-world object detection** In response to the closed-world assumption, the field has seen a pivot towards open-world object detection [20, 23, 8, 10, 25, 43, 16, 21, 36, 17], which focuses on incrementally learning new objects [20, 43, 16] in the given data. However, unknown objects might be so different to the original data that generalizing inside the given data is not enough. Moreover, prior work often adopted two stage object detectors such as Faster-RCNN [31], while some more recent work changed to transformer-based architectures such as DETR [4]. However, both architectures are not suitable for current real-time systems. The work [23] extended FCOS [35] (a one-stage anchor-free object detector) for open-world object detection, using the localization quality scores initially proposed by [21]. Similar to [23], we aim at real-time solutions. We propose a novel objectness measure via occupancy prediction rather than using localization quality measures, as the latter is biased towards the known objects.

## 3 Method

Our method, **UNCOVER**, is designed to equip real-time object detection systems with awareness of unknown objects. Figure 2 illustrates the three main aspects of UNCOVER. The model contains one additional class and an occupancy prediction head, and is trained with a mixture of AD data and other domain data for improved object diversity at training. At inference time (Phase II), the newly learned occupancy prediction serves as a measure of objectness, improving the recall of unknown objects. Given that AD systems often provide depth information for a scene, we also propose a simple and interpretable depth-based filtering for false positive detection reduction, i.e. Phase III.

### 3.1 Preliminary

We showcase our design on top of modern, one-stage, anchor-free object detectors due to their real-time capability and competitive performance. Taking YOLOX [14] and YOLOv8 [19] as examples: each has a backbone and neck network followed by two decoupled heads, i.e., one for classification and one for regression. The classification loss  $L_{cls}$  is based on binary cross-entropy and averaged

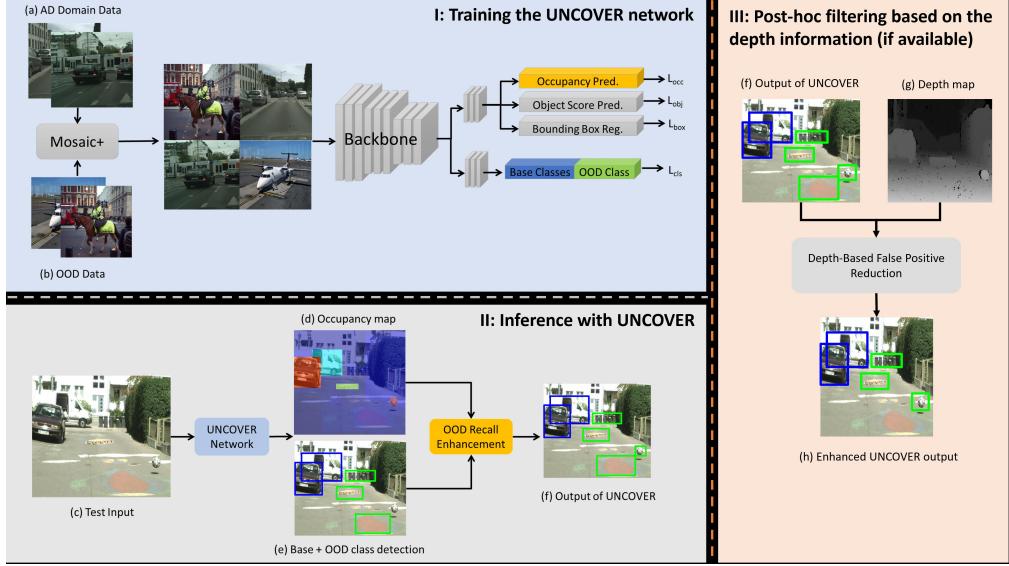


Figure 2: UNCOVER – We enable unknown object detection by adding OOD data (via Mosaic+), one extra class for OOD classification (dark blue), and one regression output (yellow) to predict the occupancy of each detection, i.e. phase I at training. In phase II, UNCOVER exploits the occupancy prediction to improve the recall of unknown objects in addition to the OOD class detection via the classification head. If depth information, commonly found in AD systems, is available, UNCOVER uses an interpretable filtering step to reduce false positive detections (see phase III). Note, in addition to Mosaic+, we also use Mixup; it is left out of the diagram for simplicity.

over all classes. Besides the regression loss  $L_{box}$  for learning the bounding box coordinates, YOLOX has an additional prediction output noted as  $Obj.$  score in the regression branch. It classifies each detection based on if it has a matched ground-truth bounding box; its training loss  $L_{obj}$  is binary cross entropy based. We describe our modification on top of this base architecture.

### 3.2 Unknown Object Detection (Phase I and II)

To achieve high recall of OOD objects, UNCOVER 1) leverages data from other domains with an OOD class, 2) trains an occupancy prediction and 3) filters based on the occupancy score.

#### 3.2.1 Extra OOD Class and Mosaic+

UNCOVER introduces an extra class, termed OOD class, so that detections can be classified as "Unknown". As the training set does not involve any OOD annotations, we include other datasets [26, 15] that have annotated objects beyond the training classes of the AD dataset. Note, there are also objects having the same semantic classes as the AD dataset. They will still be trained with the corresponding known class in the classification head. To seamlessly incorporate new data for training, we extend Mosaic, a strong data augmentation scheme initially introduced in YOLOv4 [2]. Mosaic concatenates multiple images into one, greatly improving the training data diversity. Our modification is to fetch images from two different sources instead of one, i.e., Mosaic+ as shown in Figure 2. Using data from different domains introduces a domain gap, as highlighted by [40], which can affect the efficacy of using external datasets like MS COCO[26] for training anomaly segmentation in AD contexts. To bridge this gap, we employ Mixup after Mosaic+, blending the composed image with an AD scene image, leveraging techniques also used in YOLOX [14].

#### 3.2.2 Occupancy Prediction

While MS COCO [26] and LVIS [15] have many more object classes than AD datasets, it is still unavoidable that the OOD class overfits to those specific classes in both datasets. Therefore, we introduce an occupancy prediction in the regression branch, which is supervised to evaluate the objectness in a class-agnostic manner. Then, we can decide whether to keep detections that have high

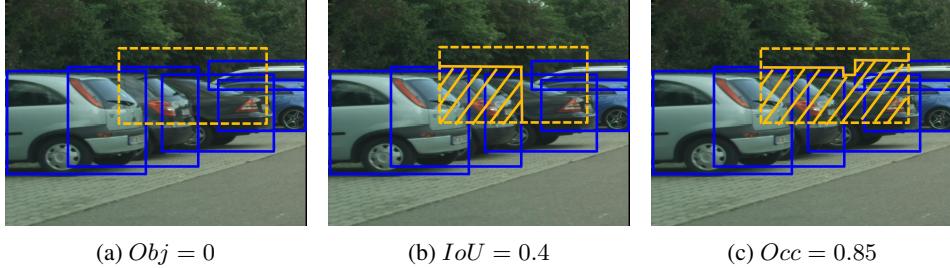


Figure 3: We compare three different objectness measures for ground truth (GT) bounding boxes (blue) and the predicted bounding box (dashed line, yellow). The following scores are the target for optimization, not the actual output. For (a)  $Obj$  [14], the score is one if positively matched to one GT box, or zero like in the presented case. For (b)  $IoU$  [21] the highest  $IoU$  achieved with one of the GT boxes. For (c) our proposed  $Occ$  concerns the intersection with all GT boxes and thus does not rely on a valid matching, like a). Thus, even when the localization is difficult, occupancy with one or more objects is easily determinable. Moreover,  $Obj$  and  $IoU$  may require matching classes, while we are class-agnostic, allowing better generalizability to unknown objects.

scores even if their classification confidences may be low. For measuring objectness, IoU predictions are commonly used [21, 18]. In the context of YOLOX, that could be the  $Obj$ . score [14]. Both scores were developed for measuring the localization quality and may not respond well to unknown objects on which the model is expected to deliver lower localization quality.

Instead of using a detection quality measure as the proxy for objectness, we introduce an occupancy prediction score. The score measures the ratio of the predicted area that overlaps with the ground truth bounding boxes. Specifically, the training loss  $L_{occ}$  based on the binary cross entropy is formulated as:

$$L_{occ}(b_{occ}) = -t_{occ} \cdot \ln(b_{occ}) - (1 - t_{occ}) \cdot \ln(1 - b_{occ}) \\ t_{occ} = \frac{|b_{pred} \cap (\bigcup_{i=0}^n b_{gt_i})|}{|b_{pred}|}, \quad (1)$$

where  $t_{occ}$  is the target and  $b_{occ}$  is the occupancy prediction. Here,  $|\cdot|$  denotes the area in pixels,  $b_{pred}$  the predicted bounding box,  $\bigcup_{i=0}^n b_{gt_i}$  is the union of all ground-truth bounding boxes ( $b_{gt_i}$ ) in the image. If that area is mostly covered by objects, the target  $t_{occ}$  gets closer to one. We also introduce an approximation for simplifying the computation of  $t_{occ}$  in the supp. material.

As illustrated in Figure 3, the predicted bounding box (yellow dashed line) contains objects, so that the anchor point generating it shall contain object features, i.e., vehicles. The IoU and  $Obj$ . score suppress their response to the object features in such a case. In contrast, the supervision signal for the occupancy prediction is stronger when the overlapping area is larger. A larger area indicates that the features yielding the bounding box prediction are mostly from objects.

### 3.2.3 OOD Recall Enhancement

After training, UNCOVER can generate detections with multi-class labels, including the OOD class. During inference, a standard filtering is applied to remove low-quality predictions which usually do not contain any objects of interest. In YOLOX, such filtering is based on the product of the maximum classification probability and the  $Obj$ . score, i.e.,  $sco$ . Only detections with  $sco \geq \mu_{sco}$  are kept. As it is challenging for the model to achieve similar localization qualities for unknown objects as for known ones, we introduce the second filtering step for OOD recall enhancement. It retains the detections with  $(sco < \mu_{sco})$  yet with a high occupancy score  $occ \geq \mu_{occ}$ . These retained ones will be considered as OOD objects in addition to those kept in the initial filtering step as OOD.

### 3.3 Depth Based False Positive Reduction (Phase III)

Monocular inputs are cost-effective but can lead RGB-based detection to depend heavily on appearance cues, which may not be always reliable. Figure 4 illustrates failure cases where non-objects are mistakenly identified due to their distinct textures from the background. Such false positives,

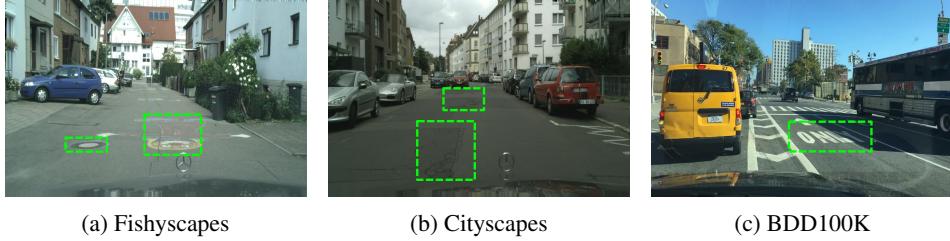


Figure 4: Examples where visual cues are not robust, yielding false positive detections.

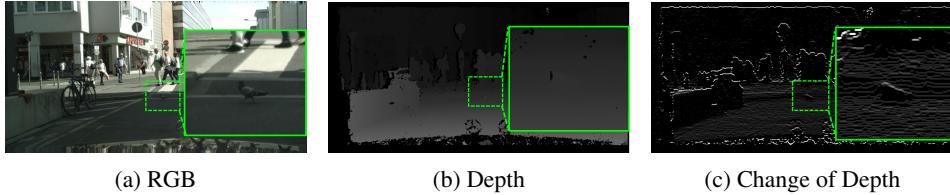


Figure 5: Geometric cues from depth. For objects with geometric shapes such as the bird, it can be detected via the change of depth in the area. For road marking, there is no depth change. Therefore, depth can be exploited to filter non-objects in near range.

particularly on drivable surfaces near the vehicle, could critically undermine the system reliability e.g. of emergency brakes. To suppress such failures, we hypothesize the most effective solution is to leverage geometric cues, which provide a more holistic view and complement the appearance cues. To verify our hypothesis and given the fact that depth information is often available in AD systems, we experiment on depth-based filtering. Note, here we only exploit the depth information for filtering the detections that are already generated by UNCOVER.

The main idea is to measure the changes in depth within each bounding box. See for example the bird in Figure 5, which is difficult to spot in the raw depth estimation but easier to discern in the change of depth. An indication of potential objects is having few local depth changes within the bounding box; on the other hand, the flat surface of the road exhibits continuous local depth changes. To detect such patterns, we devise Algorithm 1, which is low complexity, interpretable, and purely post-hoc, without the need for retraining the network. Specifically, our algorithm involves pre-processing the depth map to highlight areas with depth changes by leveraging morphological transformations [11] and the Sobel operator [33]. The algorithm determines the likelihood of an object sticking out vertically from the ground by further evaluating the proportion ( $c$ ) of pixels within a predicted bounding box that have depth change less than ten. Objects are characterized by having a few local changes. Therefore, the ones with  $c \geq \mu$  are kept. While depth estimation quality becomes worse farther away from the camera, this does not affect us negatively. Since depth change can only be detected within a certain distance, the detections in far distance have  $c$  close to its maximum and are always kept. The filtering mostly affects close objects. This is of practical interest, as closer objects are more relevant to the action for the very next step.

## 4 Experiments

**Datasets** We consider two standard AD object detection datasets for training, i.e., Cityscapes (CS) [7] and BDD100k [38]. The former encompasses images of urban street scenes in Europe, while the latter was collected in U.S.. They both have eight annotated object classes for training: person, rider, car, truck, bicycle, train, bus, and motorcycle. To facilitate learning to detect unknowns, we incorporate also images from COCO [26] and LVIS [15]. Their label space alignment with CS and BDD100k is detailed in the supplementary material.

For evaluation, we construct the unknown object annotations by improving the annotations of CS on the class 'dynamic', 'trailer', and 'caravan'. The 'dynamic' class is described as 'Things that might not be there anymore the next day/hour/minute: Movable trash bin, buggy, bag, wheelchair, ani-

---

**Algorithm 1** Depth Based False Positive Reduction

---

**Require:** Depth Image  $D$ , Predicted Bounding Box  $x_1, x_2, y_1, y_2$

# Calculation of depth change  $C$  from  $D$

$D \leftarrow dilation(D, kernel = 10)$   $\triangleright$  Morphological Closing with dilation and erosion

$D \leftarrow erosion(D, kernel = 10)$

$C \leftarrow sobel(D, direction = y, kernel = 5)$   $\triangleright$  Sobel operation in y-direction

# Calculate acceptance / rejection of every predicted bounding box  $x_1, x_2, y_1, y_2$

$bbox \leftarrow C[x_1 : y_1, x_2 : y_2]$   $\triangleright$  Pixels of bounding box in  $C$

$c \leftarrow count(bbox < 10) / size(bbox)$   $\triangleright$  Proportion with minimal depth change

**if**  $c \geq \mu$  **then**

    Accept bounding box as potentially hazardous

**else**

    Reject bounding box as non-hazardous

**end if**

---

mal<sup>1</sup>, with examples in the appendix. We derived bounding box information from semantic masks using blob detection for the non-instance-specific 'dynamic' annotation, with details in the supplementary material. BDD100k provides object detections for the classes 'traffic sign' and 'traffic light' which we do not use during training and thus can be used as unknowns during evaluation.

Besides CS and BDD100k, we also benchmark on FishyScapes Lost and Found (FS L&F) [1] and Segment Me If You Can (Anomaly / Obstacle Track)[5]. They have real OOD scenarios, but without bounding box annotations. Therefore, we process them to generate the annotations for benchmarking unknown object detection methods, and refer to the supplementary material for details.

**Evaluation Metrics** We consider three metrics: mean Average Precision (mAP in %) [26], Recall (R@100 in %) and False Positive Rate (FPR@100 in %) at the IoU threshold 0.5. mAP is measured on the training classes, i.e., knowns. R@100 and FPR@100 measure the success rate of unknown detection, where the top 100 unknown detections are kept for evaluation. Except the objects from the training classes, unknown objects are not exhaustively annotated in the datasets; therefore we cannot directly measure mAP for them. FPR@100 is computed for a specific region of interest (RoI) where we have exhaustive annotations for every pixel. In our case, we use the "road" mask to derive the RoI and report FPR within that specific region. For run-time evaluation we measure the frames per second (FPS), see details in the supplementary material.

**Implementation Details** We adopted the YOLOX architecture with a CSPNetX backbone, configured to process images at a resolution of  $640 \times 640$  pixels, as our base objection detection model [14]. Our implementation is based on the mmdetection framework<sup>2</sup> and their default parameters. The experiments were mainly conducted on one NVIDIA V100 with an average training time of 20 hours per experiment. The Mosaic+ augmentation strategy replaces two out of four images from an auxiliary OOD dataset like MS COCO and LVIS [15, 26]. Our occupancy loss is added to the original training loss of YOLOX with the weighting  $w_o = 1.0$ . To address the imbalance among Cityscapes/BDD100k classes, we applied class weights of  $[0.5, 0.9, 0.4, 1, 1, 1, 1, 0.7, 10]$ , where the last entry for the extra OOD class is up-weighted due to the greater learning complexity. For thresholding, we used grid-searched optimized values of  $\mu = 0.3$ , and  $\mu_{occ} = 0.01$  where  $\mu_{sco} = 0.01$  is the default choice in YOLOX. For more details, see the supplementary material.

#### 4.1 Unknown Object Discovery via Occupancy Prediction

To filter generated bounding boxes, state-of-the-art object detectors rely on objectness measures. Namely, a bounding box with a high objectness measure is likely to capture an object. In order to improve the recall of unknown objects, it is important to learn a generic sense of objectness, avoiding biases towards the training classes. Table 1 compares three different objectness measures, where the base model is the same, i.e., YOLOX with Mosaic+. *Obj.* score is a built-in prediction of

<sup>1</sup><https://www.cityscapes-dataset.com/dataset-overview/>

<sup>2</sup><https://github.com/open-mmlab/mmdetection>

Table 1: Comparison of objectness measures for open-world object detection using UNCOVER trained on CS [7] with OOD data from LVIS [15]. Best results are in bold.

Objectness measure	Cityscapes		BDD100K		FS L&F		Anomaly		Obstacle	
	mAP ↑	R@100 ↑	mAP ↑	R@100 ↑	R@100 ↑	R@100 ↑	R@100 ↑	R@100 ↑	R@100 ↑	R@100 ↑
<i>Obj.</i> score	34.42	13.21	12.40	39.38	48.62	<b>93.75</b>	75.56			
IoU score	35.83	11.70	11.51	39.32	52.49	<b>93.75</b>	73.33			
Occ. score (Ours)	<b>35.91</b>	<b>15.71</b>	<b>14.11</b>	<b>39.42</b>	<b>58.56</b>	<b>93.75</b>	<b>77.78</b>			

YOLOX, and we further use it to do OOD recall enhancement. As *Obj.* score also considers the class information (during SimOTA), it performs worse than the others, as the objectness measure should be class-agnostic, i.e., generalizable across knowns and unknowns. Compared to the IoU score which measures the localization quality, our occupancy measure is more responsive to unknowns, yielding higher recall across all benchmarks. Among the five benchmarks, Anomaly and Obstacle are less challenging as the objects are salient in the scenes. In contrast, the unknown objects in FS and BDD100k are rather small, e.g., lost cargo objects, and traffic signs. The lowest recalls are on the reserved unknown classes from Cityscapes. As shown in the supp. material, the objects are often small and occluded in the background without clear view, thus more challenging to detect. We refer to the supplementary material for more visual results on comparing the objectness scores. The effectiveness of OOD class and objectness score are also ablated there.

## 4.2 Comparison with Anomaly Segmentation

While segmentation and object detection are two perception tasks with different goals, we find it interesting to compare them for unknown detection. Firstly, anomaly segmentation in AD has received more attention than object detection. Secondly, both Fishyscapes [1] and SegmentMeIfYouCan [5] are anomaly segmentation benchmarks. It is expected that a segmentation model can outperform our model but at the cost of real-time condition. Table 2 compares UNCOVER with two competitive anomaly segmentation methods PEBAL [34] and RPL [28]. As they were trained on Cityscapes and use MS COCO as OOD data, we also switch from using LVIS to MS COCO for a fair comparison. The predicted mask of PEBAL and RPL are converted into bounding boxes with blob detection as elaborated in the supplementary material. Our recall performance is better than PEBAL while similar to RPL. The main differences are on Fishyscapes and Anomaly based on the visuals in supplementary. We hypothesize that the anomaly mask conversion yields smaller detections, which is helpful on FS with more smaller objects on the horizon, while hindering on Anomaly with large objects. A clear advantage of UNCOVER is its much lower complexity, achieving ten times higher throughput than PEBAL and RPL while having similar or better performance.

We also compare with open-world object detection and open-vocabulary object detection methods in the supplementary material. Unlike anomaly segmentation, they were not proposed and tailored for AD domains. The open-world object detector PROB [43] and OW-DETR [16] do not already perform well on Cityscapes. Grounding DINO [27, 42] (GDINO) and YOLO-World [6] are open-vocabulary methods that allow to detect a wide range of object classes with text prompts. We prompt large language models (LLMs) to generate potential object classes that may appear in driving scenes (but outside the AD training classes) and use them to detect potential unknown objects. GDINO performs often within 10% of UNCOVER while not being real-time, while YOLO-World achieves real-time but has a lower Recall on all 5 datasets. We conclude that our solution UNCOVER excels at detecting unknown objects in real-time while preserving known class performance.

## 4.3 Depth-based False Positive Reduction (DFR)

Our last experiments highlight the role of depth information in reducing false positives. Despite UNCOVER’s relative low FPRs (see supplementary), false positives can be further reduced using geometric cues, such as depth information. For Cityscapes and Fishycapes stereo images are available. While the focus of our work is on monocular object detection, Table 3 shows that depth-map based filtering improves all prior methods, including UNCOVER (ours). Not only does the filtering reduce FPRs, but it also improves recall. Filtering out ghost detections prevents them from consuming the budget, i.e., the top 100 detections, for recall evaluation. We see that on average DFR

Table 2: Comparison of the anomaly segmentation methods PEBAL and RPL. All methods are trained on Cityscapes [7] where MS COCO [26] are exploited as OOD data. Best number is bold and 2nd best within 50% margin is italic.

Method	Runtime		Cityscapes		FS L&F	Anomaly	Obstacle
	FPS $\uparrow$	mAP $\uparrow$	R@100 $\uparrow$				
PEBAL [34]	2.80	N/A	10.54	44.75	12.50	53.33	
RPL [28]	3.26	N/A	<b>20.80</b>	<b>70.17</b>	18.75	<b>84.44</b>	
<b>UNCOVER</b>	<b>26.29</b>	<b>35.94</b>	<i>15.62</i>	49.72	<b>100.00</b>	82.22	

Table 3: Results for Depth Based False Positive Reduction (DFR) – Relative changes between without and with DFR are given in brackets. Improvements over 20% for FPR@100 and 1% for R@100 are marked bold.

Dataset	Cityscapes		FS L&F		
	Method	FPR@100 $\downarrow$	R@100 $\uparrow$	FPR@100 $\downarrow$	R@100 $\uparrow$
PEBAL		3.9	10.54	2.8	44.75
+ DFR		3.9 (+0.0%)	11.16 ( <b>+5.9%</b> )	2.0 (- <b>28.6%</b> )	45.86 ( <b>+2.5%</b> )
RPL		7.1	20.80	9.6	70.17
+ DFR		6.5 (-8.5%)	20.80 (+0.0%)	7.6 (- <b>20.8%</b> )	69.06 (-1.6%)
YOLOWorld		8.8	12.23	60.8	33.70
+ DFR		9.1 (+3.4%)	13.30 ( <b>+8.7%</b> )	46.0 (- <b>24.3%</b> )	35.36 ( <b>+4.9%</b> )
GDINO		2.5	30.54	8.8	74.03
+ DFR		1.9 (- <b>24.0%</b> )	30.54 (+0.0%)	5.4 (- <b>38.6%</b> )	74.03 (+0.0%)
UNCOVER (CS)		6.0	15.71	11.1	58.56
+ DFR		4.6 (- <b>23.3%</b> )	15.71 (+0.0%)	8.3 (- <b>25.2%</b> )	58.56 (+0.0%)
UNCOVER (BDD)		2.2	13.93	0.5	55.25
+ DFR		2.5 (+13.6%)	17.86 ( <b>+28.2%</b> )	2.8 (- <b>44.0%</b> )	55.80 ( <b>+1.0%</b> )

reduces FPR@100 by 18.4% and increases R@100 by 4.1%. Since stereo depth estimation is not always available we evaluate the impact of monocular depth estimation in the supplementary.

## 5 Broader Impact & Limitations

Autonomous driving has a major potential impact on our society. However, reliable and fast system especially in unexpected situations are key for save driving. Thus, we have focused on enabling existing real-time models with simple modifications. Our final performance is then naturally also limited by the base model. Our method can be applied with a new model, and a joint design may boost the performance further. Moreover, while our focus is not on data augmentation, exposing models to some OOD data during training has been an effective solution. However, not every possible real-world object has an equal chance to be traffic relevant, e.g., many objects from the “accessory” category in MS COCO used in this study. Thus, prioritizing objects that are more likely in AD scenes but hard to collect in real world is an interesting direction. We see no direct harmful application of our proposed method for autonomous driving.

## 6 Conclusion

In this work, we present a novel method UNCOVER (unknown class object detection for autonomous vehicles in real-time). By integrating a novel occupancy prediction head and augmenting training on the OOD class with diverse datasets, our method achieves state-of-the-art performance on five different datasets under the contrast of real-time prediction, with up to 25% improved recall. Additionally, our depth-based filtering technique, utilizing geometric cues, significantly reduces false positives by 18.4% and improves recall by 4.1%. This shows the effectiveness of our methods on addressing open-world AD challenges in real-time.

## References

- [1] Blum, H., Sarlin, P.E., Nieto, J., Siegwart, R., Cadena, C.: The FishyScapes Benchmark: Measuring Blind Spots in Semantic Segmentation. *International Journal of Computer Vision* **129**(11), 3119–3135 (2021). <https://doi.org/10.1007/s11263-021-01511-6>
- [2] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934* (2020)
- [3] Bochkovskiy, A., Wang, C.Y., Liao, H.Y.M.: Yolov4: Optimal speed and accuracy of object detection (2020)
- [4] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
- [5] Chan, R., Lis, K., Uhlemeyer, S., Blum, H., Honari, S., Siegwart, R., Fua, P., Salzmann, M., Rottmann, M.: SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation (NeurIPS), 1–13 (2021), <http://arxiv.org/abs/2104.14812>
- [6] Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y.: YOLO-World: Real-Time Open-Vocabulary Object Detection (2024), <http://arxiv.org/abs/2401.17270>
- [7] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) **2016-Decem**, 3213–3223 (2016)
- [8] Dhamija, A., Gunther, M., Ventura, J., Boult, T.: The overlooked elephant of object detection: Open set. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1021–1030 (2020)
- [9] Du, X., Gozum, G., Ming, Y., Li, Y.: Siren: Shaping representations for detecting out-of-distribution objects. In: Advances in Neural Information Processing Systems. vol. 35, pp. 20434–20449 (2022)
- [10] Du, X., Wang, Z., Cai, M., Li, Y.: Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197* (2022)
- [11] Fisher, R., Perkins, S., Walker, A., Wolfart, E.: Hypermedia image processing reference. England: John Wiley & Sons Ltd pp. 118–130 (1996)
- [12] Galessos, S., Argus, M., Brox, T.: Far away in the deep space: Dense nearest-neighbor-based out-of-distribution detection appendix. ICCV (2023)
- [13] Gao, C., Wang, G., Shi, W., Wang, Z., Chen, Y.: Autonomous driving security: State of the art and challenges. *IEEE Internet of Things Journal* **9**(10), 7572–7595 (2021)
- [14] Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430* (jul 2021), <http://arxiv.org/abs/2107.08430>
- [15] Gupta, A., Dollar, P., Girshick, R.: Lvis: A dataset for large vocabulary instance segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition **2019-June**, 5351–5359 (2019). <https://doi.org/10.1109/CVPR.2019.00550>
- [16] Gupta, A., Narayan, S., Joseph, K.J., Khan, S., Shahbaz Khan, F., Shah, M., Khan, F.S., Shah, M., Shahbaz Khan, F., Shah, M., Khan, F.S., Shah, M., Shahbaz Khan, F., Shah, M., Khan, F.S., Shah, M.: OW-DETR: Open-world Detection Transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9235–9244 (2022), <https://github.com/akshitac8/OW-DETR>.
- [17] Huang, H., Geiger, A., Zhang, D.: GOOD: Exploring Geometric Cues for Detecting Objects in an Open World (2022)
- [18] Huang, H., Geiger, A., Zhang, D.: GOOD: Exploring geometric cues for detecting objects in an open world. In: The Eleventh International Conference on Learning Representations (ICLR) (2023), <https://openreview.net/forum?id=W-nZDQyuy8D>
- [19] Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (Jan 2023), <https://github.com/ultralytics/ultralytics>

- [20] Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5830–5840 (2021)
- [21] Kim, D., Lin, T.Y., Angelova, A., Kweon, I.S., Kuo, W.: Learning Open-World Object Proposals without Learning to Classify. *IEEE Robotics and Automation Letters* **7**, 5453–5460 (2021)
- [22] Kim, D., Ka, W., Ahn, P., Joo, D., Chun, S., Kim, J.: Global-Local Path Networks for Monocular Depth Estimation with Vertical CutDepth (jan 2022), <http://arxiv.org/abs/2201.07436>
- [23] Konan, S., Liang, K.J., Yin, L.: Extending One-Stage Detection with Open-World Proposals. arXiv preprint arXiv:2201.02302 (jan 2022), <http://arxiv.org/abs/2201.02302>
- [24] Konolige, K., Agrawal, M., Bolles, R.C., Cowan, C., Fischler, M., Gerkey, B.: Outdoor mapping and navigation using stereo vision. In: Experimental Robotics: The 10th International Symposium on Experimental Robotics. pp. 179–190. Springer (2008)
- [25] Liang, W., Xue, F., Liu, Y., Zhong, G., Ming, A.: Unknown Sniffer for Object Detection: Don't Turn a Blind Eye to Unknown Objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3230–3239 (2023), <https://github.com>
- [26] Lin, T.y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. European conference on computer vision pp. 740–755 (2014)
- [27] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J.J.J., Li, C., Yang, J.J.J., Su, H., Zhu, J., Zhang, L., Others, Zhang, L., Others: Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. arXiv preprint arXiv:2303.05499 (mar 2023), <http://arxiv.org/abs/2303.05499>
- [28] Liu, Y., Ding, C., Tian, Y., Pang, G., Belagiannis, V., Reid, I., Carneiro, G.: Residual Pattern Learning for Pixel-wise Out-of-Distribution Detection in Semantic Segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1151–1161 (nov 2023), <http://arxiv.org/abs/2211.14512>
- [29] OpenAI: Introducing ChatGPT. Blog Entry (2022)
- [30] Rai, S.N., Cermelli, F., Fontanel, D., Masone, C., Caputo, B.: Unmasking anomalies in road-scene segmentation. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 4037–4046 (jul 2023), <http://arxiv.org/abs/2307.13316>
- [31] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE transactions on pattern analysis and machine intelligence* **39**(6), 1137–1149 (2015)
- [32] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images Lecture Notes in Computer Science. ECCV'12: Proceedings of the 12th European conference on Computer Vision - Volume Part V Part V(Chapter 54), 746–760 (2012)
- [33] Sobel, I., Feldman, G., Others: A 3x3 isotropic gradient operator for image processing. a talk at the Stanford Artificial Project in pp. 271–272 (1968)
- [34] Tian, Y., Liu, Y., Pang, G., Liu, F., Chen, Y., Carneiro, G.: Pixel-wise energy-biased abstention learning for anomaly segmentation on complex urban driving scenes. In: European Conference on Computer Vision. pp. 246–263 (nov 2022), <http://arxiv.org/abs/2111.12264>
- [35] Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully Convolutional One-Stage Object Detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9627–9636 (apr 2019), <http://arxiv.org/abs/1904.01355>
- [36] Wang, Y., Yue, Z., Hua, X.S., Zhang, H.: Random Boxes Are Open-world Object Detectors. Proceedings of the IEEE/CVF International Conference on Computer Vision pp. 6233–6243 (2023), <https://github.com/scuwyh2000/RandBox>.
- [37] Wu, D., Liao, M.W., Zhang, W.T., Wang, X.G., Bai, X., Cheng, W.Q., Liu, W.Y.: Yolop: You only look once for panoptic driving perception. *Machine Intelligence Research* pp. 1–13 (2022)

- [38] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 2633–2642 (2020). <https://doi.org/10.1109/CVPR42600.2020.00271>
- [39] Zhan, J., Luo, Y., Guo, C., Wu, Y., Meng, J., Liu, J.: Yolopx: Anchor-free multi-task learning network for panoptic driving perception. Pattern Recognition **148**, 110152 (2024). <https://doi.org/10.1016/j.patcog.2023.110152>, <https://www.sciencedirect.com/science/article/pii/S003132032300849X>
- [40] Zhang, D., Sakmann, K., Beluch, W., Huttmacher, R., Li, Y.: Anomaly-aware semantic segmentation via style-aligned ood augmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 4065–4073 (October 2023)
- [41] Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018), <https://openreview.net/forum?id=r1Ddp1-Rb>
- [42] Zhao, X., Chen, Y., Xu, S., Li, X., Wang, X., Li, Y., Huang, H.: An Open and Comprehensive Pipeline for Unified Object Grounding and Detection (2024), <http://arxiv.org/abs/2401.02361>
- [43] Zohar, O., Wang, K.C., Yeung, S.: Prob: Probabilistic objectness for open world object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11444–11453 (2023)

## Appendix / supplemental material

The supplementary material of the main paper is structured as follows:

- In Appendix A, more details about our method UNCOVER.
  - A.1 a low-complexity approximation of our occupancy prediction loss function.
  - A.2 hyper-parameter selection for OOD recall enhancement.
  - A.3 hyper-parameter selection for depth based false positive reduction.
- In Appendix B, more experimental details.
  - B.1 Dataset setting and processing for benchmarking unknown object detection.
  - B.2 Runtime measurement and comparison.
  - B.3 Comparison with anomaly segmentation models.
  - B.4 Comparison with open-vocabulary models.
- In Appendix C, additional experiment results
  - C.1 Comparison results with two open-vocabulary object detection models.
  - C.2 Comparison results with two transformer-based open-world object detection models.
  - C.3 Ablation of the impact of the different parts of UNCOVER.
  - C.4 Comparison results with monocular depth estimation for DFR.
  - C.5 Object detection visualizations generated by our model (UNCOVER), YoloWorld [6], GDINO [42] and RPL [28]. Additionally visual results of YoloPX [39].
  - C.6 Visuals of occupancy vs. object predictions across datasets.
  - C.7 Full reproducibility results for Figure 7.

## A More implementation Details of UNCOVER

### A.1 Approximation of the Loss Function for Occupancy Prediction

To speed up the loss computation and gradient computation, we resort to an approximation of the loss in Equation 1, used for training the proposed occupancy prediction. Specifically, the intersection between the predicted bounding box and the union of all ground-truth boxes is approximated by its upper bound

$$\begin{aligned} |b_{pred} \cap (\bigcup_{i=0}^n b_{gt_i})| &= |\bigcup_{i=0}^n (b_{pred} \cap b_{gt_i})| \\ &\leq \sum_{i=0}^n |(b_{pred} \cap b_{gt_i})|. \end{aligned}$$

As an upper bound, the approximation potentially overestimates the intersection. It becomes tight when there are no or very few occlusions among the ground-truth bounding boxes. Empirically, we verified that this approximation works as effectively as the original loss formulation, while also reducing the training time.

### A.2 Ablation on the filtering mechanism for OOD recall enhancement

One of our core contributions is the introduction of the occupancy score and using it to improve the recall of OOD objects. While the training loss aims at learning generic objectness, the learned occupancy score can be used in multiple ways. Aiming at real-time processing, we favor simple threshold-based solutions, namely, filtering detections based on real value comparison to the occupancy score. Through our ablation study, we find that the final performance is not sensitive to the exact threshold value  $\mu_{occ}$  when it is in a reasonable region. We use the same value as that used

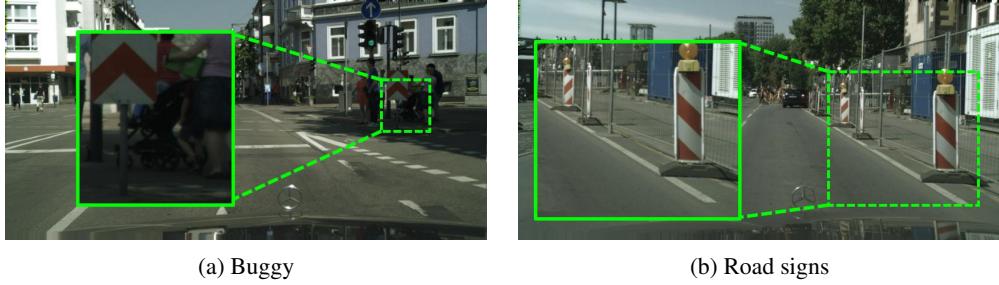


Figure 6: Two examples of unknowns in Cityscapes from the 'dynamic' class

by YOLOX [14] for filtering objects from the training classes, i.e.,  $\mu_{occ} = \mu_{sco} = 0.01$ . It is kept the same for all tested benchmarks. In subsection C.6 we show multiple visual examples that highlight how occupancy is more salient for objects than the  $obj$  metric of [14]. Especially for highly occluded or difficult instances, our occupancy prediction can more robustly identify the existence of such objects.

### A.3 Ablation on the filtering mechanism for depth-based FPR reduction

To further reduce the false positive rate, we use filtering with a single threshold setting  $\mu$ . As shown in Figure 7 (for numbers see Table 7 and Table 8), the performance is not sensitive to  $\mu$ . Our selection is based on a grid search on a hold-out validation set. In general, one can analyze the depth changes of the annotated training class objects and derive the threshold, as such geometry cues are rather class-agnostic. Moreover, we only need to verify whether the bounding box contains any object; thus this process is not sensitive to slight depth changes.

## B More Experiment Details

### B.1 Datasets and evaluation

As we use MS COCO [26] and LVIS [15] as auxiliary data to augment the training on the AD datasets, Cityscapes [7] and BDD100k [38], their label spaces must be aligned; some classes from COCO and LVIS share the same semantic concepts as the training classes of Cityscapes and BDD100k, including for instance cars and persons. Other classes do not share the same semantic concepts of the 8 training classes and we merge them into an 'OOD' class.

We further note that while 'traffic sign' and 'traffic light' are common training classes for AD, Cityscapes only provides semantic instead of instance masks for these. Thus we removed them from the known classes during training, and used them for OOD recall and false positive evaluation. Furthermore, we use the 'dynamics' class in Cityscapes for OOD evaluation. Two examples are given in Figure 6. For the classes for which only semantic masks are available, we extracted bounding boxes from the semantic masks with blob detection using the Python version of OpenCV's 'findContours'. In principle this could have led to either merging or splitting of instances depending on the connectivity of the mask. We found this not to be an issue in practice since most samples are individual objects which do not overlap, or are separated by other classes.

For FPR@100 evaluation, we focus on the road area, as this is most relevant to driving safety. Moreover, it is important that in the region of interest we need pixel-wise annotations to ensure that there are no un-annotated objects. For the object detection benchmark of BDD100k, the ground-truth road mask annotation is only available for a subset of the training set. Therefore, for our Cityscapes trained model, we use the BDD100k training set to evaluate FPR. When training on BDD100k, we do not report FPR numbers due to the absence of road masks in the validation set. For Segement Me If You Can, we use the provided in-distribution masks to determine the ROI for FPR calculation. For the obstacle track this corresponds to the street surface as well. For the Anomaly Track, everything which is in-distribution for the original semantic segmentation mask is marked as a region of interest.

For obtaining the depth information, we tried both stereo inputs and monocular depth estimator, i.e., a Global-Local Path Network [22] specifically fine-tuned on the NYUv2 dataset [32]. We mainly

report results based on stereo depth estimation [24] but show in subsection C.4 also results for the described off-the-shelf monocular depth estimator.

## B.2 Runtime evaluation

We focus in our evaluation on comparing magnitudes of differences in runtime. Overall, our runtime evaluation aims at ranking each model from the perspective of run-time capability; the exact computation cost and hardware complexity are beyond the scope of the evaluation. The run time evaluation was conducted on a NVIDIA V100 and a NVIDIA RTX2070. We calibrated the performance difference between the used hardware and the reported FPS is based on averaged run-times across 100 repetitions. Since we evaluated based on reported system times, it can be expected that the numbers slightly change based on the remaining load of the system. This is not an issue for the reported results since the major differences are a magnitude slower or faster.

## B.3 Conversion of anomaly semantic masks into bounding boxes

The evaluation metrics commonly used by the anomaly segmentation methods in the literature are threshold free. However, for practical usage, the per-pixel OOD scores generated by the anomaly segmentation methods need to be mapped into a binary mask, classifying each pixel as either OOD or not OOD. As the methods in the literature do not provide a way of setting such thresholds, we devise the following strategy to generate (instance-wise) OOD detections based on the per-pixel OOD scores. We try multiple thresholds and fuse their results. Specifically, for each threshold we determine which parts of the anomaly scores are above the threshold and use blob detection on the thresholded image. The score for each detected blob / bounding box is the used threshold, since it describes the lowest confidence in the anomaly scores for a connected patch. Thus, when using lower thresholds more parts of the image are considered to be unknown and larger bounding boxes are predicted. Visual examples for converted RPL [28] detections are shown in subsection C.5.

## B.4 Open Vocabulary Prompting

When using open-vocabulary models (such as GDINO and YOLO-World) for OOD-aware object detection, we generate text-based prompts in the following manner. For the known classes, we directly use the class names, i.e., 'person', 'rider', 'car', 'truck', 'bus', 'train', 'motorcycle', 'bicycle'. For unknown classes, it is impossible to be exhaustive. Nevertheless, it is possible to query LLMs such as ChatGPT [29] to generate the most AD-relevant classes beyond the training classes. Moreover, we make use of the definition of 'dynamic' class provided in Cityscapes, as they are actual objects which have already been observed in AD scenes. The final prompting texts for unknown objects include 'traffic sign', 'traffic light', 'buggy', 'road obstacle', 'construction sign', 'wheel chair', 'animal', 'trash bin', 'wheel chair', 'pallet', 'wheel', 'baggage', 'traffic cone', 'box', 'branch', 'bicycle', 'ball', 'toy', 'dog', 'bird', 'skateboard', 'scooter', 'cat', 'stroller', 'bench', 'fence', 'puddle', 'pothole', 'manhole cover', 'flower pot', 'bollard', 'roadwork barrier', 'fallen sign', 'shopping cart', 'ladder', 'sandbag', 'construction equipment', 'temporary road sign', 'street art installation', 'lost clothing', 'spilled cargo', 'advertising board', 'fire hydrant', 'electric scooter', 'table', 'chair', 'sun shade', 'helicopter', 'airplane', which are used in addition to the 8 training class names in above.

Note, 'traffic sign' and 'traffic light' are treated as unknowns for the reason we provided in subsection B.1. Moreover, the class names are not mutually exclusive, e.g., animal and dog are both in the name list. This is because the open-vocabulary model reacts to each text prompt differently. Both animal and dog can become a traffic participant, creating hazardous road situations. In our setup, predicting either of them leads to an identical predicted label, i.e., 'OOD' object.

## C More results

### C.1 Comparison with open-vocabulary object detection

VLMs enable detecting unseen unknowns by specifying names of potential traffic-relevant objects beyond training classes, allowing open-vocabulary detectors to identify them at inference. Grounding DINO (GDINO) [27, 42] demonstrated a strong generalization across different benchmarks. YOLO-World [6] is a very recent work with the real-time capability. By querying GPT [29], we

Table 4: Comparison with the open-vocabulary object detection models. Both GDINO [42] and YOLOWorld [6] are trained on large-scale object detection data and demonstrate strong zero-shot generalization capabilities cross benchmarks. Our model is respectively trained with Cityscapes [7] and BDD100k [38] together with LVIS [15] as the OOD data. Best is marked bold and 2nd best is italic.

Method	Runtime	Cityscapes		BDD100K		FS L&F	Anomaly	Obstacle
	FPS ↑	mAP ↑	R@100 ↑	mAP ↑	R@100 ↑	R@100 ↑	R@100 ↑	R@100 ↑
GDINO [42]	6.05	32.76	<i>30.54</i>	24.58	78.41	<b>74.03</b>	<b>100.00</b>	<b>93.33</b>
YOLOWorld [6]	<b>30.59</b>	22.14	12.23	21.44	29.24	33.70	<b>100.00</b>	62.22
UNCOVER (CS)	26.29	<b>35.91</b>	15.71	14.11	39.42	58.56	93.75	77.78
UNCOVER (BDD)	26.29	25.14	17.86	<b>30.33</b>	<b>84.35</b>	55.80	<b>100.00</b>	88.89

Table 5: Comparison with the open world object detection models, i.e., PROB [43] and OW-DETR [16], with our proposed UNCOVER. All of them are trained on Cityscapes [7].

Dataset	Cityscapes		BDD100K		FS L&F	Anomaly	Obstacle	Runtime
	Method	mAP ↑	R@100 ↑	mAP ↑	R@100 ↑	R@100 ↑	R@100 ↑	FPS ↑
PROB	23.06	10.54	7.36	0.57	4.42	56.25	6.67	14.81
OW-DETR	18.67	4.91	6.54	0.48	4.97	25.00	6.67	15.27
UNCOVER (CS)	35.91	15.71	14.11	39.42	58.56	93.75	77.78	26.29

generated 57 highly traffic relevant object classes beyond the training classes, e.g., buggy, trash bin or animal (details in supp. material) for GDINO and YOLO-World. In Table 4, we observe that ours outperforms YOLO-World with regard to all recalls, e.g., up to 25% improved recall on FS. The real-time capability is on a par. The more complex and thus slower GDINO surpasses our method in most cases but we are often at least within 10% of their result showing similar performance.

We have also shown UNCOVER trained on both Cityscapes and BDD100k, which have a domain gap. The generalization from BDD100k to Cityscapes is much better than the opposite direction due to the larger size of BDD100k. Due to exposure to very diverse data at pre-training, the open-vocabulary models also generalize better than UNCOVER (CS) on BDD100k.

## C.2 Comparison with Transformer-based Open-world Object Detectors

The recent trend in open-world object detection moves towards using transformer-based architectures. These models are typically trained with different losses and configurations than the CNN based ones. While transformers have great potential, and display advantages over CNN-based ones, they are currently still comparatively less data efficient at training and more expensive at inference time.

In Table 5, we report additional results with models such as PROB [43] and OW-DETR [16], which are much slower than YOLOX and YOLO-World. On the performance side, they are also inferior. We hypothesize that their transformer-based architectures require much more training data than Cityscapes. So far, their strong open-world detection performance was demonstrated on object detection benchmarks such as MS COCO, which has a lot more training samples than Cityscapes.

## C.3 Ablation study on occupancy prediction vs. OOD classification

We introduce two architecture modifications for UNCOVER, i.e., extra OOD class and occupancy prediction. From Table 6, we can observe that removing the classification head leads to a severe drop in the recall. The reduction of the FPR is a symptom of less detected objects overall.

Removing occupancy prediction for OOD recall enhancement compromises both FPR and recall. These observations indicate that both components contribute to the success of UNCOVER. It is also worth noting that UNCOVER preserves the mAPs on the known classes, compared to the base model YOLOX. Based on Table 6 and Table 2, it is more beneficial to use LVIS over MS COCO as the data source for OOD-awareness training. LVIS has a much larger object diversity, which is

Table 6: Ablation of the OOD classification and occupancy prediction for unknown object detection. Models are trained on Cityscapes [7] with LVIS [15] as OOD data. The much reduced FP without OOD class is due to reduced OOD detections in general, as we can see from the number drops on Recall.

Method	Cityscapes			FS L&F	
	mAP↑	FPR@100↓	R@100↑	FPR@100↓	R@100↑
YOLOX [14]	36.03	-	-	-	-
UNCOVER	35.91	<b>6.0</b>	<b>15.71</b>	<b>12.6</b>	<b>58.56</b>
- OOD class.	<b>36.11</b>	<b>0.3</b>	13.57	<b>0.8</b>	48.07
- Occ. pred.	35.92	21.3	8.39	20.9	55.80

CS (Mono) CS (Stereo) FS (Mono) FS (Stereo) FPR@100 R@100

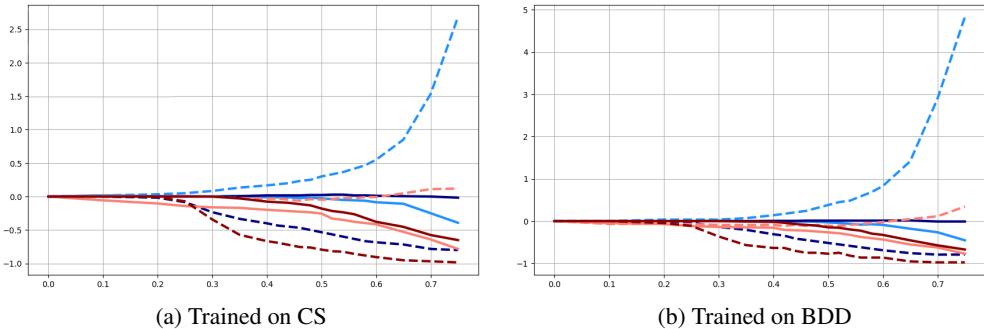


Figure 7: *Relative* changes in FPR@100 and R@100 (Y-Axis) when applying different thresholds ( $\mu$  at X-Axis) for depth-based filtering on top of UNCOVER – Lower depth estimation accuracy (the monocular depth estimator [22] in this case vs. stereo depth estimation) leads to increased sensitivity in determining the threshold  $\mu$ .

a key factor for learning generic sense of objectness. More details and results can be found in the supplementary material.

#### C.4 Depth-based False Positive Redcution with Monocular Depth Estimation

As stereo inputs are not always available, we further study the influence of depth estimation quality on the performance. For comparison, we take an off-the-shelf depth estimator Global-Local Path Network [22] to generate a depth map, expected to be of lower quality than the one obtained from the stereo inputs. From Figure 7, we can observe the benefit of high-quality depth information includes less sensitivity to the threshold setting  $\mu$  in Algorithm 1, and more pronounced gains in FPR and Recall. A lower quality depth map leads to a more erratic and premature performance decline as thresholds.

#### C.5 Qualitative Evaluation

We report more visual results in Figure 8, Figure 9, Figure 10, Figure 11, Figure 12 and Figure 13 for our method UNCOVER, YOLOWorld [6], GDINO [42] and RPL [28]. The results are best viewed digitally. We see that our method highlights unknown objects with high confidence while preserving predictions of known classes.

As a comparison we also provide in Figure 14 a visual comparison to Panoptic Segmentation [39] under the real-time constraint. Since no OOD or anomaly detection is integrated into the segmentation task, OOD objects are randomly detected or not on the driveable surface. The images also highlight why a direct comparison is not possible since objects like the giraffe are only partially on the driveable area and thus not fully segmented.



Figure 8: Comparison of different methods for OOD-aware object detection on Fishyscapes. UNCOVER detects unknown objects, while maintaining high known class prediction accuracy. At the chosen threshold YOLOWorld correctly detects the OOD object, but misses known classes. GDINO and RPL (used here for object detection) suffer from many false positives.

### C.6 Occupancy visualization

We also include three comparisons between  $obj$  and our  $occ$  prediction for all datasets in Figure 15, Figure 16, Figure 17, Figure 18 and Figure 19. We see that  $occ$  provides a more reliable indication of the objects, including both unknowns (e.g., ball in Figure 16a)) and knowns (e.g. group of persons in Figure 15a). In contrast, the  $obj$  score as a measure for localization quality has trouble indicating objects when the localization is not of high quality, e.g., a group of persons due to high occlusion or unknown objects due to lack of direct supervision. However, for those objects, it is safer to detect them as objects rather than ignore them as background.



Figure 9: Similar observations to Figure 8. It is also interesting to note that FishyScapes only annotates unknown objects on the road, i.e., lost cargos. However, many detections made by UNCOVER in the background are actually also objects, even if they are neither part of the training classes nor of the labeled OOD objects in FishyScapes. Therefore, we evaluate average recalls on unknown object detection. Average precision would penalize the right object detections due to lack of exhaustive object annotations.

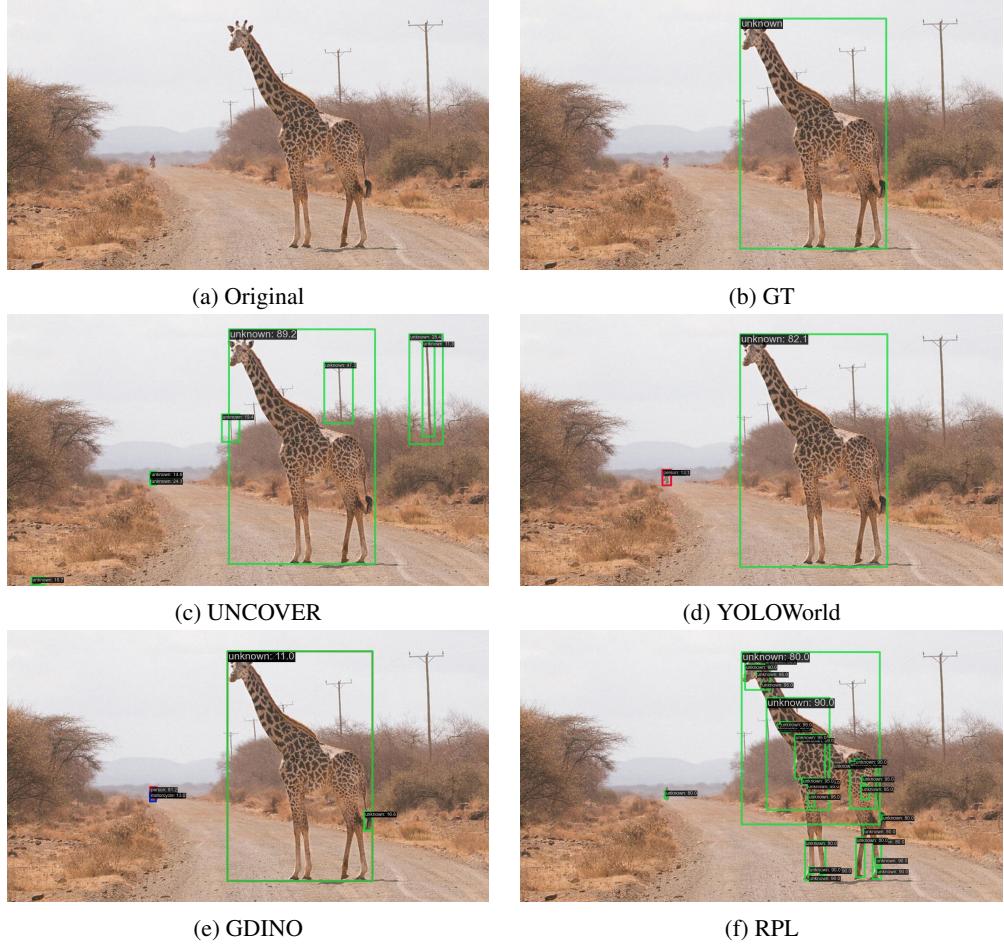


Figure 10: Anomaly: In this case, the giraffe is a salient object which is rather easily detected by all models. RPL has many small detections with poor localization quality. In contrast, UNCOVER detects background objects well in addition to the giraffe, even though they are not annotated.



Figure 11: Anomaly: In this example, both GDINO and UNCOVER have a ghost detection, i.e., tree shadow, whereas YOLOWorld has less positive detections on the background objects. RPL still suffers from many small detections. To remove the ghost detection, our proposed depth-based filtering can be applied.

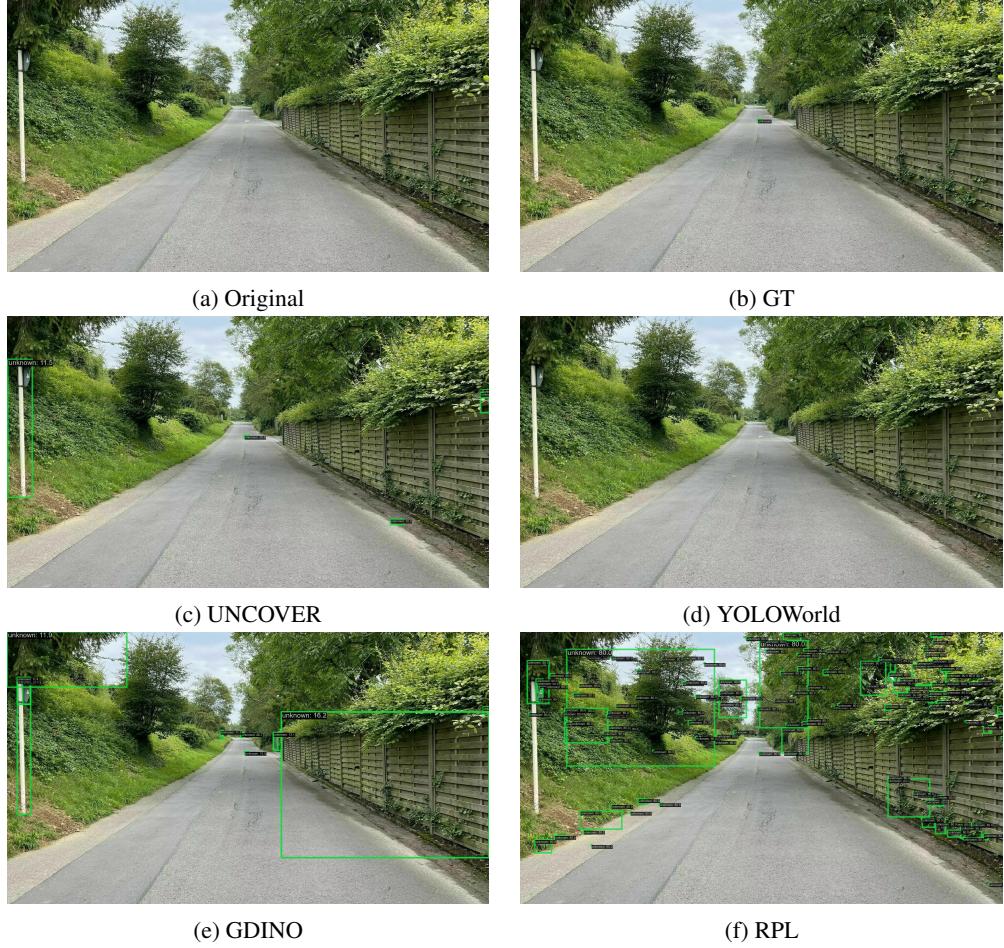


Figure 12: Obstacle: this example demonstrates that UNCOVER is also capable at detecting further away small objects, that are either missed by YOLOWorld or detected at a much larger false positive rate, as shown for RPL.



Figure 13: Obstacle: Similar to the previous example. However, YOLOWorld detects the largest object while not picking up the others.

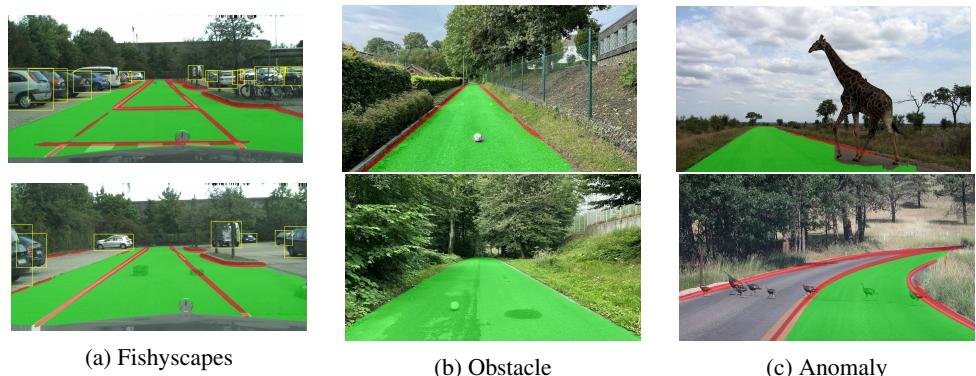


Figure 14: Visual results of YoloPX [39] across three different anomaly segmentation tasks – We see that the drivable area is sometimes aware of OOD objects like the ball and the giraffe. However it is not even consistent for the ball on the obstacle dataset.

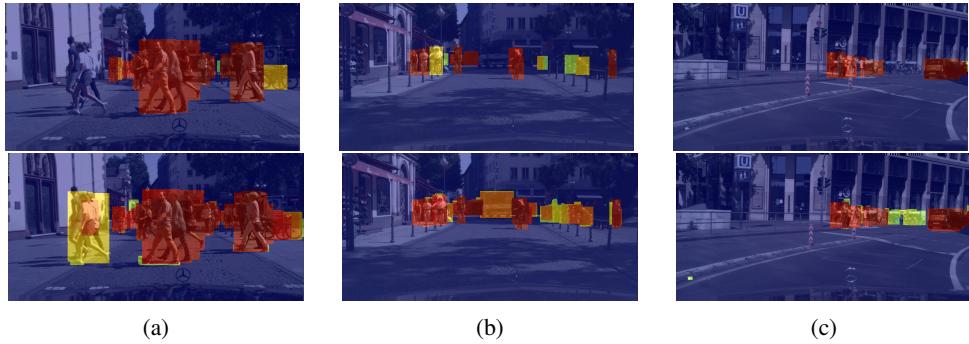


Figure 15: Comparison of  $obj$  (Top) vs. our  $occ$  prediction (Bottom) on the Cityscapes dataset. The  $occ$  score detects several objects that are not captured by the  $obj$  score.

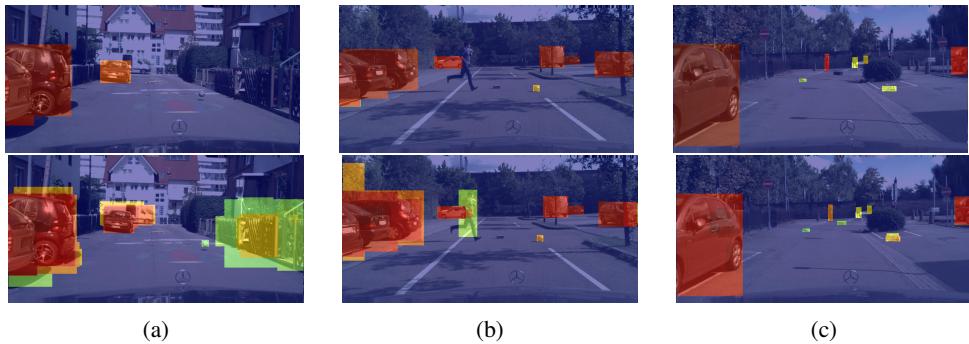


Figure 16: Comparison of  $obj$  (Top) vs. our  $occ$  prediction (Bottom) on Fishyscapes dataset

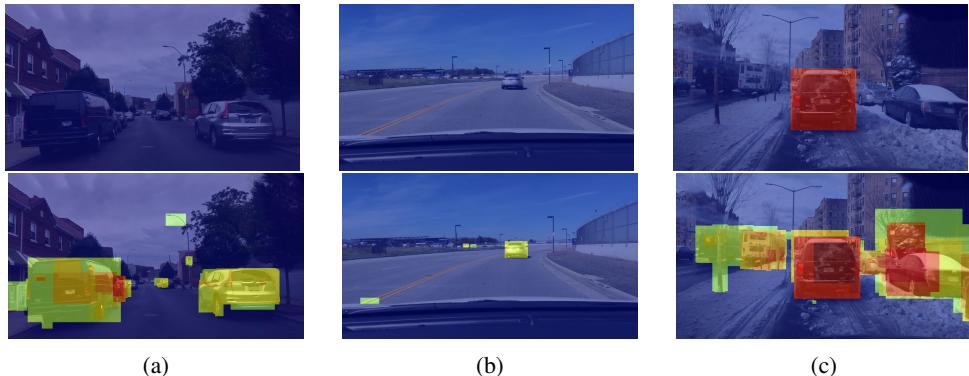


Figure 17: Comparison of  $obj$  (Top) vs. our  $occ$  prediction (Bottom) on BDD100K dataset

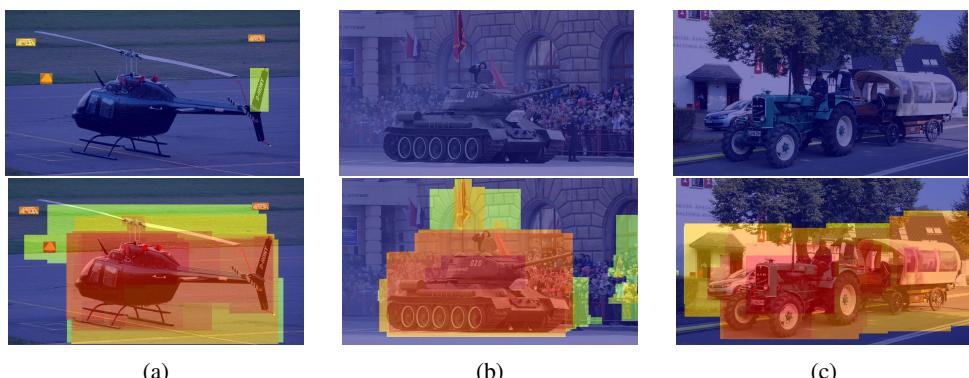


Figure 18: Comparison of  $obj$  (Top) vs. our  $occ$  prediction (Bottom) on Anomaly dataset

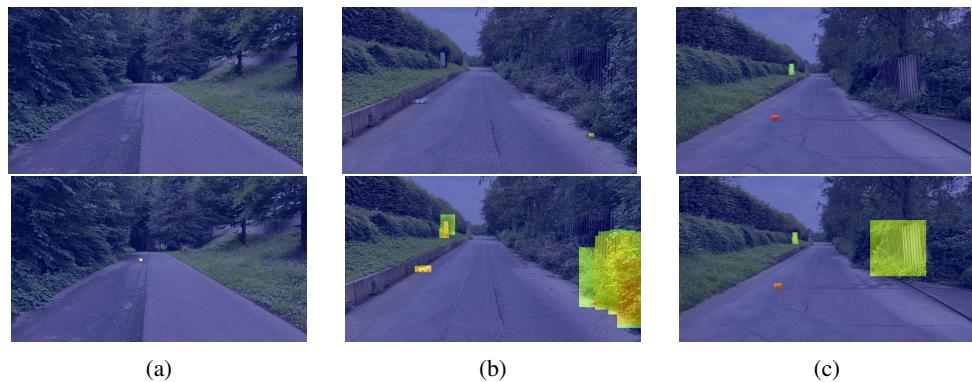


Figure 19: Comparison of *obj* (Top) vs. our *occ* prediction (Bottom) on Obstacle dataset

### C.7 Reproducibility of Figure 7

The full results for the presented Figure 7 are given in Table 7 and Table 8.

Table 7: Results for Figure 7a for reproducibility

Dataset	Cityscapes				FS L&F				
	Depth	Stereo		Monocular		Stereo		Monocular	
		FPR@100	R@100	FPR@100	R@100	FPR@100	R@100	FPR@100	R@100
$\mu = 0$	0.60	15.71	0.60	15.71	1.26	58.56	1.26	58.56	
$\mu = 0.1$	0.61	15.98	0.60	15.71	1.27	55.25	1.26	58.56	
$\mu = 0.2$	0.62	15.80	0.59	15.71	1.26	52.49	1.25	58.56	
$\mu = 0.25$	0.63	15.80	0.55	15.71	1.27	50.28	1.17	58.56	
$\mu = 0.3$	0.65	15.71	0.46	15.71	1.26	49.17	0.83	58.56	
$\mu = 0.35$	0.68	15.80	0.40	15.80	1.23	48.62	0.54	56.91	
$\mu = 0.4$	0.70	15.54	0.36	15.98	1.22	46.96	0.42	54.14	
$\mu = 0.42$	0.71	15.54	0.34	15.98	1.22	46.41	0.39	53.59	
$\mu = 0.44$	0.72	15.54	0.33	15.98	1.20	45.86	0.35	53.04	
$\mu = 0.46$	0.74	15.36	0.32	15.98	1.18	45.30	0.31	51.93	
$\mu = 0.48$	0.75	15.45	0.30	16.07	1.21	44.75	0.30	50.83	
$\mu = 0.5$	0.78	15.27	0.28	16.07	1.20	43.65	0.26	48.07	
$\mu = 0.52$	0.80	15.09	0.26	16.16	1.21	39.23	0.23	45.86	
$\mu = 0.54$	0.82	14.91	0.24	16.16	1.24	38.12	0.22	44.75	
$\mu = 0.56$	0.85	14.82	0.22	15.98	1.23	36.46	0.18	43.09	
$\mu = 0.58$	0.88	14.73	0.20	15.98	1.25	35.36	0.15	39.78	
$\mu = 0.6$	0.93	14.37	0.19	15.89	1.25	34.25	0.12	36.46	
$\mu = 0.65$	1.11	14.02	0.17	15.80	1.32	27.62	0.06	32.04	
$\mu = 0.7$	1.52	11.79	0.13	15.71	1.40	20.99	0.04	24.86	
$\mu = 0.75$	2.21	9.55	0.12	15.45	1.41	12.71	0.02	20.44	

Table 8: Results for Figure 7b for reproducibility

Dataset	Cityscapes				FS L&F				
	Depth	Stereo		Monocular		Stereo		Monocular	
		FPR@100	R@100	FPR@100	R@100	FPR@100	R@100	FPR@100	R@100
$\mu = 0$	0.29	17.86	0.29	17.86	0.44	55.80	0.44	55.80	
$\mu = 0.1$	0.29	17.86	0.29	17.86	0.41	53.04	0.44	55.80	
$\mu = 0.2$	0.30	17.86	0.29	17.86	0.41	51.93	0.42	55.80	
$\mu = 0.25$	0.30	17.86	0.28	17.86	0.41	49.17	0.39	55.80	
$\mu = 0.3$	0.30	18.04	0.25	17.86	0.40	48.07	0.28	55.80	
$\mu = 0.35$	0.31	17.77	0.23	17.86	0.40	47.51	0.19	55.25	
$\mu = 0.4$	0.33	17.68	0.20	17.95	0.40	46.96	0.16	54.70	
$\mu = 0.42$	0.34	17.59	0.19	17.95	0.39	44.20	0.16	53.59	
$\mu = 0.44$	0.35	17.68	0.17	17.95	0.40	43.65	0.13	51.93	
$\mu = 0.46$	0.36	17.68	0.16	18.04	0.39	43.09	0.11	50.83	
$\mu = 0.48$	0.38	17.59	0.15	18.04	0.40	41.99	0.11	49.72	
$\mu = 0.5$	0.40	17.32	0.14	18.04	0.39	40.88	0.10	47.51	
$\mu = 0.52$	0.42	17.05	0.13	18.04	0.39	39.78	0.11	46.96	
$\mu = 0.54$	0.43	16.70	0.12	18.04	0.40	37.57	0.08	44.75	
$\mu = 0.56$	0.46	16.34	0.11	18.04	0.41	34.81	0.06	43.09	
$\mu = 0.58$	0.49	16.52	0.10	18.04	0.42	33.15	0.06	39.23	
$\mu = 0.6$	0.53	16.25	0.09	18.04	0.43	31.49	0.06	37.57	
$\mu = 0.65$	0.70	14.64	0.07	18.12	0.46	24.86	0.02	30.39	
$\mu = 0.7$	1.13	13.13	0.06	17.68	0.49	20.99	0.01	23.76	
$\mu = 0.75$	1.69	9.73	0.06	17.68	0.59	12.71	0.01	18.23	