OUT-OF-DISTRIBUTION OBJECT DETECTION IN STREET SCENES VIA SYNTHETIC OUTLIER EXPOSURE AND TRANSFER LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Out-of-distribution (OOD) objects are often silently overlooked by object detectors, which produce no detection for them, leaving downstream modules with nothing to act on and no indication that anything was missed. A reliable object detector must detect OOD objects by localizing and correctly classifying them as OOD. Existing approaches primarily focus on OOD detection for image classification or rely on auxiliary branches, and typically do not treat in-distribution (ID) and OOD objects in a unified manner. We present **SynOE-OD**, a **Syn**thetic **O**utlier-**E**xposure-based **O**bject **D**etection method, which uses text-guided inpainting and an open-vocabulary object detector to generate and verify synthetic outliers offline. These are then used to fine-tune an object detector of choice with an added OOD class, which requires no auxiliary branch at inference time. On the OoDIS benchmark, SynOE-OD outperforms the strongest open-vocabulary baseline, reaching $AP_{50..95}$ of $57.6\,\%$, $29.4\,\%$ and $39.2\,\%$ on RoadObstacle, RoadAnomaly and Fishyscapes, against 48.5, 22.7 and 22.7. This state-of-the-art performance holds for both fine-tuned closed-set and open-vocabulary detectors while maintaining competitive ID performance in street scenes.

1 INTRODUCTION

Classical deep object detectors are typically trained on a fixed set of classes, and objects that appear outside this label space are assigned to the background Dhamija et al. (2020). Hence, object detectors explicitly learn not to detect unlabeled object classes, in particular classes that are semantically out-of-distribution (OOD). In safety-critical applications, e.g., autonomous driving Henriksson et al. (2023) or medicine Gutbrod et al. (2025), an object that is never localized leaves the downstream system nothing to act on. Detecting OOD objects matters as much as detecting in-distribution (ID) objects Bogdoll et al. (2022); Shoeb et al. (2025).

A significant amount of work addresses OOD detection for image classification Hendrycks & Gimpel (2016); Liang et al. (2017); Lee et al. (2018); Sun et al. (2022); Ren et al. (2019); Liu et al. (2020); Oberdiek et al. (2022) and semantic segmentation Chan et al. (2021a); Zhao et al. (2024); Rai et al. (2023); Blum et al. (2019); Bevandić et al. (2019), while OOD object detection remains comparatively understudied Du et al. (2022b); Ilyas et al. (2024). Most methods for OOD object detection Du et al. (2022b;a); Wilson et al. (2023) train a detector on ID data and add an auxiliary branch that scores each predicted box as ID or OOD. Such methods then inherit the same blind spot of classical object detectors, since the score only applies to boxes the detector predicts. OOD objects are thus recognized only if the detector happens to localize them, and their localization is neither learned nor evaluated. OoDIS Nekrasov et al. (2025) evaluates the detection of anomalies at the instance level in street scenes, which makes it possible to measure OOD object detection performance.

Outlier Exposure (OE) Hendrycks et al. (2018) exposes a model to auxiliary outlier data so that OOD inputs receive high uncertainty scores. VOS Du et al. (2022b) adapts this idea to detection by synthesizing virtual outliers in feature space. Its evaluation follows the Pascal VOC/COCO setting Everingham et al. (2010); Lin et al. (2014), where ID and OOD objects do not coexist in

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

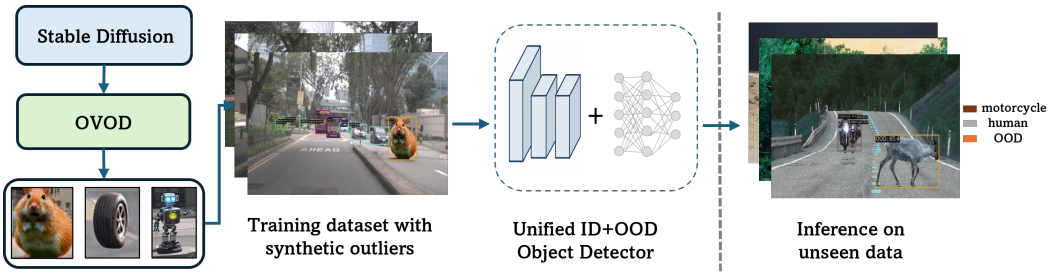

Figure 1: Overview of the SynOE-OD method, where an object detector is fine-tuned to jointly detect ID and OOD objects. NuImages is the training set augmented with synthetic outliers. At inference time, the detector is evaluated on an unseen real-world test set, e.g., RoadAnomaly.

the same image. Recent advances in image generation via Stable Diffusion Rombach et al. (2022); RunwayML (2023) make OE more scalable, as outliers can now be directly inserted into ID scenes.

We therefore present SynOE-OD, illustrated in fig. 1, which incorporates synthetic outliers directly into the image space and fine-tunes object detectors to jointly detect ID and OOD objects. Unlike methods that score the ID detector's predicted boxes as ID or OOD, SynOE-OD supervises outliers as detection targets in the same manner as any ID class, and thereby learns to detect OOD objects that would otherwise be assigned to the background. Specifically, Stable Diffusion inpaints outliers at plausible locations in street scenes by replacing ID objects or filling free road regions, while Grounding DINO (GDINO) Liu et al. (2024b) annotates them with precise boxes and accurate labels. An object detector of choice is then fine-tuned via transfer learning with an additional $(n{+}1)$-st OOD class, where n denotes the number of ID classes. This formulation matches the requirements of applications such as autonomous driving, where known and unknown objects must be localized and classified in the same scene. In practice, OE is known to be sensitive to the choice of outlier data Wang et al. (2023b); Zhang et al. (2023); its performance degrades when the training outliers do not resemble the test-time anomalies. We study this dependence for synthetic outliers in the object detection setting for street scenes.

Our key contributions are as follows:

- We introduce SynOE-OD, which extends closed-set and open-vocabulary object detectors to jointly detect ID and OOD objects, requiring neither real OOD data nor an auxiliary branch at inference time.
- SynOE-OD sets a new state-of-the-art on OoDIS and generalizes to real unseen road obstacles, while retaining ID performance close to that of an ID-only detector across the three detector configurations we evaluate.
- We analyze what makes synthetic outliers effective for OOD object detection, and show that performance is more sensitive to the quality of outlier annotations than to their amount and semantic content.

2 Related Work

Outlier exposure and OOD detection. Post-hoc scoring methods for closed-set classifiers Hendrycks & Gimpel (2016); Liang et al. (2017); Lee et al. (2018); Sun et al. (2022); Ren et al. (2019); Liu et al. (2020) rank the predictions a model already made. OE Hendrycks et al. (2018) has numerous extensions in classification Tao et al. (2023); Li & Vasconcelos (2020); Zhang et al. (2023), Yu et al. (2024); Wang et al. (2023a); Jiang et al. (2024); Liu & Zach (2024). Among image-level OOD detection methods, Dream-OOD Du et al. (2023) and Li et al. (2025) learn a text-conditioned latent space and decode outliers from it as whole images. All of these decide whether an input image is anomalous, not where the anomaly is. Anomaly segmentation extends image-level OOD detection to a dense prediction task with established benchmarks, such as Fishyscapes Blum et al. (2019) and SegmentMeIfYouCan Chan et al. (2021a). Early approaches scored pixels by uncertainty estimation or entropy maximization Oberdiek et al. (2020); Chan et al. (2021b). Recent methods Loiseau et al. (2024); de Jorge et al. (2024) inpaint outliers with a diffusion model,

UGainS Nekrasov et al. (2023) obtains anomaly instances from pixel-level uncertainty, where supervision and evaluation remain at the pixel level, which does not transfer automatically to object detection. A dense predictor assigns a score to every pixel, so an anomaly always receives a prediction, whereas object detectors produce sparse predictions and may silently fail on unknown objects.

OOD object detection. Detecting unknown objects at the instance level is shared with open-world object detection (OWOD) Joseph et al. (2021), which incrementally learns unknowns it encounters Gupta et al. (2022); Zohar et al. (2023), whereas OOD object detection operates on a closed-set label space and detects objects that fall outside it as unknown. Most methods in this setting modify the detector with an auxiliary uncertainty branch Du et al. (2022b), a generalized objectness score Liang et al. (2023), a distance-based embedding head Du et al. (2022a) or features from sensitivity-aware layers, trained on perturbed outliers Wilson et al. (2023). These share a component that must be designed for OOD detection training and retained at inference time. VOS Du et al. (2022b) samples virtual outliers from low-density regions of class-conditional Gaussians fitted to ID box features and passes them through the classification head, so they supervise how a detected box is classified, not whether it is proposed in the first place. SyncOOD Liu et al. (2024a) inpaints outliers and trains a binary classifier on box features of an off-the-shelf detector. In all of these, OOD detection applies to boxes the base detector already proposed, the OOD decision is made by an added scoring component and overlooked objects are not recovered. Alternatively, zero-shot open-vocabulary object detectors (ZS-OVOD) Ilyas et al. (2024) localize many street-scene obstacles Nekrasov et al. (2025), without OOD-specific training, however, their output is prompt-dependent and an unknown object is by definition not known ahead of time. Dream-Box Isaac-Medina & Breckon (2025) instead trains the region proposal network of Faster R-CNN on outliers, while its classification and box regression heads remain ID, and argues, that treating OOD objects as an additional class may not generalize. That formulation is nevertheless established in image classification Vernekar et al. (2019) and semantic segmentation Bevandić et al. (2022). Unlike these methods, we treat outliers as detection targets that receive the same localization and classification supervision as any ID class, and find that the additional class transfers to real OOD objects.

3 METHOD

We propose SynOE-OD, that adapts an object detector to detect OOD and ID objects jointly. We formulate the OOD object detection problem in section 3.1. In section 3.2, section 3.3 and fig. 2 we describe (i) the synthetic outlier generation procedure based on Stable Diffusion for a given dataset and (ii) the annotations for synthetic outliers using GDINO. Lastly, we give details on the transfer-learning to integrate the synthetic outliers during fine-tuning in section 3.4.

3.1 PROBLEM STATEMENT

For unified ID and OOD object detection, we consider a set of labeled ID objects X_{ID} from a domain of interest, where each instance is a triplet (x, b, y) of an object x, its bounding box b and a class label y from n known classes $Y_{\text{ID}} = \{y_1, \ldots, y_n\}$. The remaining objects whose class is not contained in Y_{ID} form the set X_{OOD}. They are considered OOD and receive a single additional label y_{n+1}, denoted by y_{OOD}. Given an object detector f_θ, the objective is to learn parameters θ that localize the objects of X_{ID} and classify them among the n known classes, and in the same scene localize the objects of X_{OOD} and assign them to y_{n+1} rather than to the background. The goal is to extend conventional object detection to the open-set setting.

3.2 SYNTHETIC OUTLIER GENERATION

Outlier vocabulary. To obtain atypical object classes, we use ChatGPT OpenAI (2024) to return a list of candidate objects that are unusual for a given dataset. Our application is autonomous driving perception, so we query ChatGPT to '*give a list of objects that are unusual for a street scene*'. The returned list $L_{unusual} = \{l_1, l_2, \ldots, l_k\}$ contains k classes that fall into *household items*, *animals* and other *miscellaneous* objects. The full list of outliers for street scenes is given in Appendix(section A).

Scene editing. We use Stable Diffusion RunwayML (2023) to inpaint unusual objects into street scenes, and modify two kinds of image regions. These regions are either locations of ID objects or

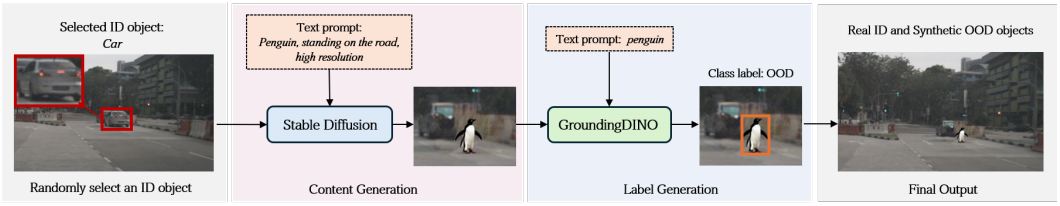

Figure 2: Overview of the synthetic outlier data generation process. The input image consists of ID objects, X_{ID}, some of which are replaced with synthetic outliers using Stable Diffusion in the content generation step. The synthetic outliers are assigned to the OOD class, y_{OOD}, with a refined bounding box, b_{OOD} using GDINO.

free-space areas on the road. For the former, we randomly select an ID object x_{ID} with bounding box b_{ID} and class label y_{ID}, $(x_{ID}, b_{ID}, y_{ID}) \in X_{ID}$. To keep the edit minimally invasive while integrating it smoothly into the scene, an enlarged image crop around the box b_{ID} is passed to the model rather than the box alone, paired with a text prompt sampled from $L_{unusual}$ returned by ChatGPT. The input text-image pair thus consists of a random class $l_{unusual} \in L_{unusual}$ and the cropped image region to generate the desired content. The generated content is then pasted back into the original image. The free-space inpainting follows the same procedure but differs in how the region is selected. The road regions are extracted from the ground truth masks, and multiple fixed-size boxes are placed in random locations. The image content inside those boxes is then modified.

3.3 Annotation Refinement

Text-guided inpainting does not always reliably generate the prompted content, a failure known as prompt neglect Manukyan et al. (2025). What failures are possible depends on the region the inpainting mask covers. In free space, the prompted object may not be generated at all, leaving the region without an object. When an ID object is replaced, the model may instead regenerate a variant of that same ID object, and the mask region is filled with an edited ID object, see fig. 3. In this case an object is generated and occupies the region, so neither a presence check nor an overlap test and similarity filter against b_{ID}, proposed in Liu et al. (2024a) catches it. Both cases if left uncorrected, introduce label noise, the first by annotating an empty region as OOD and the second by annotating an ID object as OOD.



Figure 3: Left: original scene. Right: the same scene where the ID object *car* was regenerated instead of an outlier.

We therefore employ GDINO for its exceptional zero-shot performance on semantically known classes, and distinguish three scenarios: 1) *bounding box refinement* for correctly inpainted OOD objects, 2) *label reassignment* when the inpainting reproduces the ID object it was meant to replace, and 3) *bounding box removal* for empty inpaintings. These three steps help us to obtain accurate bounding boxes and class labels, shown in fig. 2. GDINO operates on image-text pairs (B, l), where B is a crop around the original bounding box b_{ID} of the replaced ID object and l is some text prompt. We limit the input image to a crop rather than the full image, since on the full image the refinement is inconsistent and often misses the inpainted object. To estimate the quality, we manually reviewed the updated annotations for a subset of the dataset.

Bounding box refinement proceeds as follows: For each correctly inpainted OOD object the corresponding bounding box is refined by incorporating b_{ID}. We query GDINO with an image-text pair $(B_{ID}, l_{unusual})$, where B_{ID} is the image region inside the original bounding box b_{ID} and $l_{unusual} \in L_{unusual}$ is the class sampled for inpainting. The returned bounding box b_{OOD} replaces the b_{ID} and is assigned the class label $y = y_{n+1}$.

Label reassignment detects inpaintings that reproduced the original ID object and neglected the text prompt $l_{unusual} \in L_{unusual}$, we query GDINO with the outlier category $l_{unusual}$ concatenated with

the ID class label y_{ID} of the replaced object,

$$l = l_{unusual} \circ y_{ID} := l_{\text{unusual}} \cdot y_{\text{ID}}, \tag{1}$$

e.g. "penguin . car", GDINO outputs the predicted class label y:

$$y = GDINO(b_{ID}, l_{unusual} \circ y_{ID}). \tag{2}$$

If the predicted class is y_{ID}, Stable Diffusion did not follow the prompt $l_{unusual}$ but regenerated a variant of the object it was meant to replace. We revert these samples by restoring the original image content and retain the class label by setting $y = y_{ID}$. Reverted samples are not resampled, hence, the number of OOD instances is $11\,\%$ lower than the inpainting attempts.

Bounding box removal is performed in cases where GDINO returns no detections, the inpainting carries no recoverable object. We treat such inpaintings as uninformative and discard the corresponding box b_{ID}, thereby excluding it from being assigned to any specific object class.

3.4 Transfer Learning for OOD detection

To optimize ID and OOD detection jointly, we start from object detectors pre-trained on large-scale datasets, e.g., Objects365 or medium-scale datasets such as COCO, and fine-tune them via transfer learning on our inpainted datasets that combine real and synthetic objects. The real objects preserve ID performance, whereas the synthetic outliers extend the detector to semantic concepts beyond the ID classes. Following section 3.1, we add an n+1-st class bucket for OOD objects. Thereby, in our formulation the n ID classes are the set of object categories relevant to the original scenes, while objects outside this label space are recovered as OOD objects and are assigned to the n+1-st class.

4 Experiments

We describe the models, datasets and evaluation protocol in the experimental setup 4.1, the quantitative results in 4.2, detailed ablations in 4.3 and 4.4 and implementation details in Appendix A.

4.1 Experimental setup

Object Detection Models. We apply SynOE-OD on both closed- and open-vocabulary object detectors: DINO-DETR (DD) Zhang et al. (2022) which is vision-only and GDINO Liu et al. (2024b), which additionally conditions on text. Both are fine-tuned on our inpainted datasets, which contain real ID objects alongside synthetic outliers. For DINO-DETR (DD) we use two backbones, i.e., ResNet-50 He et al. (2016) and Swin-L Liu et al. (2021), each initialized from COCO-pretrained detector weights. To fine-tune GDINO we use a Swin-T Liu et al. (2021) backbone initialized from the weights pre-trained on O365, GoldG, GRIT, and V3Det.

Datasets and Evaluation. We use three street scene datasets: NuImages Caesar et al. (2020), Cityscapes Cordts et al. (2016) and BDD100K Yu et al. (2020). NuImages (NU) is our primary training set and supplies the ID classes. Cityscapes (CS) and BDD100K (BDD) are added to the training data to test whether the OOD results are particularly influenced by domain gaps between OOD and ID data. We evaluate OOD object detection on RoadObstacle (RO), RoadAnomaly (RA), Fishyscapes (FS), included in OoDIS, and on LostAndFound (LF) Pinggera et al.. For evaluating ID performance, we use the held out validation set from NuImages. Since we are specifically interested in the detection capability of object detectors, we align the evaluation with standard object detection practice Padilla et al. (2020), and adopt the evaluation protocol of OoDIS. Thus, we report COCO-style Average Precision ($AP_{50..95}$, AP_{50}, AP_{75}, $AP_{S,M,L}$). FPR95 and AUROC are standard image-level OOD detection metrics Hendrycks & Gimpel (2016); Liang et al. (2017); Yang et al. (2024) that require separate ID and OOD sets and are not applicable in our case.

Augmented Datasets for Fine-tuning. Following the generation procedure described in section 3.2, we augment the base datasets by inserting synthetic outliers, while retaining the original ID annotations, illustrated in the Appendix fig. 4. For NuImages this yields 10,702 images carrying 82,197 ID and 15,604 OOD annotations, so outliers account for $16.0\,\%$ of the labelled objects. For Cityscapes, 2,975 images with 44,621 ID and 5,568 OOD annotations, whereas for BDD100k, 2,000 images with 22,202 ID annotations and 3,109 OOD annotations. These augmented datasets are what the object detectors are fine-tuned on.

Table 1: Results on the OoDIS benchmark. Size specific AP is mentioned in the Appendix section A.

	RoadObstacle			RoadAnomaly			Fishyscapes		
Method	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}	$AP_{50:95}$	AP_{50}	AP_{75}
ZS-OVOD	48.5	77.6	52.2	22.7	33.3	23.3	22.7	33.4	24.5
UGainS	8.1	11.6	8.9	11.7	17.5	12.1	15.6	24.3	18.3
SynOE-OD (DD:ResNet50)	48.8	79.4	49.4	26.9	35.8	29.0	35.3	55.3	37.9
SynOE-OD (DD:Swin-L)	**57.6**	**87.5**	**61.7**	25.7	32.1	27.2	35.9	49.5	40.7
SynOE-OD (GDINO:Swin-T)	55.6	85.1	60.2	**29.4**	**37.3**	**31.5**	**39.2**	**55.8**	**42.2**

Baselines. We compare against three kinds of baseline, summarized in the Appendix table 9. The first are the published state-of-the-art OoDIS methods, ZS-OVOD Ilyas et al. (2024) and UGainS Nekrasov et al. (2023). The second is a training-based OOD method, VOS Du et al. (2022b), which we train on NuImages and evaluate through the same OoDIS protocol. The third is the models that differ from SynOE-OD in not having seen synthetic outliers.

4.2 Main Results

Comparison on the OoDIS benchmark. OoDIS scores anomalies at the instance level, which currently makes it the only publicly-available benchmark where localizing OOD objects is measured. We submitted all three detectors fine-tuned on inpainted NuImages and Cityscapes (table 1). SynOE-OD establishes new state-of-the-art performance on all three test sets, by over 16 percentage points (pp) on Fishyscapes, 9 pp on RoadObstacle and 7 pp on RoadAnomaly. The absolute performance across the three benchmarks is not uniform, which is expected as the datasets differ in domain and in the type of OOD objects they contain. RoadAnomaly images are collected from the web rather than recorded by a camera mounted on a vehicle, so the viewpoint and object scale are very different, which makes it furthest from our training domain. RoadObstacle and Fishyscapes have a forward-looking viewpoint consistent with NuImages and contain small OOD objects scattered across the street, the setting our free-space inpainting reproduces directly, and the numbers reflect this. On RoadObstacle, AP_{50} reaches 87.5%, so obstacles far away and in isolation are reliably detected. RoadAnomaly instead contains dense groups of adjacent anomalies, e.g., images with a flock of sheep, whereas our training images carry at most three outliers.

Comparison to trained OOD baseline. For VOS, no official results have been reported on OoDIS. For a fair comparison we retrain it on original NuImages and evaluate it under the OoDIS protocol. We also include an ID-only baseline trained on NuImages, same as VOS, whose predictions are remapped to the OOD class and evaluated class-agnostically (table 2). SynOE-OD (DD:ResNet50) is fine-tuned on inpainted NuImages and compared against these two baselines. SynOE-OD outperforms VOS on all three benchmarks. VOS detects very few OOD objects but has a competitive ID mAP, and so does the ID-only baseline. The ID-only model outperforms VOS on every benchmark and recovers far more objects in RoadAnomaly with five times as many false positives. Thus, one of these barely proposes boxes and the other proposes far too many. Supervising outliers as detection targets, as in our method, is what balances precision and recall at once.

4.3 Results across object detectors and datasets

Generality across detectors and backbones. Closed-set detectors are trained only on ID classes leaving OOD objects without an explicit detection target, causing them to be treated as background. We therefore test whether SynOE-OD supports OOD detection in closed-set detectors and also benefits OVODs. SynOE-OD (DINO-DETR), fine-tuned on inpainted NuImages with both ResNet-50 and Swin-L backbones, generalizes to real OOD objects in the test benchmarks (table 3), although the model never encounters real OOD objects during training. Swin-L performs best when the test domain closely matches the training domain (RoadObstacle) and worst were it does not (RoadAnomaly), which is consistent with a larger backbone fitting the training distribution more. The comparison for GDINO is particularly important, since its zero-shot formulation already detects unseen objects without any task-specific training, and therefore together with its prompt ensemble

Table 2: Comparison to feature-space outlier synthesis and ID-only baseline. All three methods see the same ID data during training. Recall is computed over all ground-truth OOD instances

Method	Benchmark	$AP_{50:95}$	TP	FP	FN	Recall	ID
ID-only	RoadObstacle	11.3	142	144	415	25.5	**42.2**
	RoadAnomaly	2.4	445	6772	286	**60.9**	
	LostAndFound	5.4	553	3932	1375	28.7	
VOS	RoadObstacle	1.4	16	17	541	2.9	42.0
	RoadAnomaly	2.2	291	2359	440	39.8	
	LostAndFound	2.1	334	921	1594	17.3	
SynOE-OD (ours)	RoadObstacle	**47.6**	474	471	83	**85.2**	40.6
	RoadAnomaly	**25.2**	419	1222	312	57.3	
	LostAndFound	**29.8**	1245	2565	683	**64.6**	

Table 3: OOD and ID object detection results with three detectors compared against GDINO zero-shot and prompt ensemble Ilyas et al. (2024). ID_{base} is the same detector without the OOD class and trained on original NuImages.

Method	RO	RA	LF	ID	ID_{base}
GD zero-shot	48.3	19.5	26.4	—	—
GD prompt ensemble	48.5	22.7	—	—	—
SynOE-OD (DD:R50)	48.7	26.0	**32.0**	38.8	40.4
SynOE-OD (DD:Swin-L)	**56.5**	24.5	27.0	49.1	50.8
SynOE-OD (GDINO:Swin-T)	54.3	**29.3**	29.2	45.8	47.2

serves as a strong baseline. SynOE-OD (GDINO) improves the performance by 6, 7 and 3 pp on RoadObstacle, RoadAnomaly and LostAndFound. Synthetic outliers supply a training target for OOD objects that prompting alone does not provide. Our method creates OOD detection capability in a closed-set detector and improves it in OVODs, with a small reduction of at most $\sim$2 pp in ID performance.

Effect of the text prompt. OVODs such as GDINO are conditioned on text; consequently, the name assigned to the OOD class can have a significant effect on its OOD detection performance. OOD objects cannot be enumerated beforehand, hence the class name has to be generic, and unlike prompt-based methods, we keep it fixed at inference time. GDINO encodes class names with a BERT text encoder, and we compare how it represents different class names (table 4). Specifically, we measure their cosine similarity to the embedding of *obstacle*, due to its conceptual relevance to out-of-distribution objects (table 4b); the other candidates are listed for context. The class name *OOD* carries no meaning: BERT splits it into two subword tokens and its embedding lies far from that of *obstacle*. Therefore, the model has to learn a new concept from scratch. As a result, we use *anomaly*, which improves RoadAnomaly by 5.5 pp (table 4a) while LostAndFound loses 1.7 pp, with ID performance almost unaffected. Training with *'anomaly'* also starts at a lower loss, 10.06 against 10.36, consistent with a better-initialised text embedding. Hence, a semantically meaningful class name provides a stronger initialization and improves OOD object detection.

Table 4: Effect of the OOD class name on SynOE-OD (GDINO).

(a) $AP_{50..95}$ score for GDINO fine-tuned with two different prompts with a learning rate of 2.0×10^{-4}. The ID mAP is unaffected (47.3 against 47.4).

Class name	RO	RA	LF
`OOD`	53.0	21.4	27.4
`anomaly`	53.2 (+0.2)	26.9 (+5.5)	25.7 (−1.7)

(b) BERT tokens and cosine similarity to the *obstacle* class name.

Class prompt	Tokens	cos
`OOD`	`['o','##od']`	0.21
`unknown`	`['unknown']`	0.57
`object`	`['object']`	0.71
`anomaly`	`['anomaly']`	0.74

Mitigating the domain gap with ID data variety. There is a significant domain gap between the NuImages training set and the evaluation benchmarks. We ask whether more variety in the ID training data can mitigate this gap. Cityscapes is particularly relevant to LostAndFound, as both datasets were recorded in the same country and with the same camera setup. BDD100K in contrast shares its acquisition domain with none of the benchmarks, adding it therefore tests whether generic appearance diversity is beneficial. Following the same inpainting procedure, we incorporate Cityscapes and BDD100K by only replacing ID objects, and add them both in the training set (table 5). On LostAndFound, adding Cityscapes narrows the domain gap and increases the performance for SynOE-OD by 3.3 pp (DD:Swin-L) and 4.3 pp (GDINO), and adding BDD100k adds another point for both detectors, whereas ResNet50 does not benefit. The performance on RoadObstacle and RoadAnomaly essentially stays constant. ID performance decreases by a point for each detector, which we attribute to evaluating on a validation set exclusively drawn from NuImages. Hence, SynOE-OD mitigates the domain gap mainly when the ID training data matches the target domain, while generic diversity contributes marginally.

Table 5: Effect of domain diversity in the training set. The parentheses show the difference to when trained on NuImages only.

Method	**Training data**	RO	RA	LF	NU
SynOE-OD (DD:R50)	+ CS	48.8 (+0.0)	26.1 (+0.1)	31.2 (−1.3)	37.4 (−1.4)
	+ CS + BDD	49.1 (+0.3)	26.0 (+0.0)	31.9 (−0.6)	37.7 (−1.1)
SynOE-OD (GD:Swin-T)	+ CS	55.6 (+1.3)	28.2 (−1.1)	33.5 (+4.3)	45.0 (−0.8)
	+ CS + BDD	55.0 (+0.7)	27.5 (−1.8)	34.7 (+5.5)	44.8 (−1.0)
SynOE-OD (DD:Swin-L)	+ CS	57.6 (+1.1)	25.2 (+0.7)	30.3 (+3.3)	48.5 (−0.6)
	+ CS + BDD	56.4 (−0.1)	24.4 (−0.1)	31.3 (+4.3)	48.3 (−0.8)

ID/OOD trade-off. Joint ID and OOD detection can affect ID performance in two ways: the additional OOD class competes with the ID classes for predictions, and replacing ID objects with outliers removes training annotations for the affected classes. To quantify the combined impact of these effects, we compare class-wise ID performance against a baseline without an OOD class and with all of its original ID annotations retained. As shown in table 6, SynOE-OD retains its ID mAP within 2.0 pp of the baseline. The performance reduction across classes is not uniform and is dependent on how frequently each class appears in the dataset. The underrepresented classes *motorcycle*, *bicycle* and *trailer* exhibit the largest decreases, whereas the abundant classes *car* and *pedestrian* decline only by about a point. However, *bus* is minimally affected, despite its rarity, as none of its objects are replaced during outlier generation. Nearly all of the reduction is concentrated in classes that are both rare and replaced. SynOE-OD enables joint ID and OOD object detection while preserving about 95 % of the baseline's ID performance.

Table 6: Per class ID performance in $AP_{50:95}$. The annotation count is in the Appendix table 19.

Class	ID baseline	SynOE-OD
car	58.3	57.1 (−1.2)
pedestrian	42.3	41.4 (−0.9)
truck	42.8	40.6 (−2.2)
motorcycle	47.3	44.9 (−2.4)
bus	42.4	41.7 (−0.7)
bicycle	35.8	32.3 (−3.5)
trailer	16.6	13.4 (−3.2)
mAP	40.8	38.8 (−2.0)

4.4 What makes synthetic outliers effective

Synthetic outlier vocabulary. Here we measure how far SynOE-OD generalizes to semantic concepts that are not part of its outlier vocabulary. The outlier vocabulary is proposed by an LLM (ChatGPT) without any reference to the test sets, thus, some of its classes inevitably appear in them. Out of the 90 classes in the outlier vocabulary, 20 occur in the test sets. We regenerate the training set by removing those 20 classes, so their inpaintings are undone. As a counterpart we remove 20 non-overlapping classes, which are contained in none of the test sets, i.e., vacuum, chair, lamp, printer etc. table 7. For SynOE-OD (DINO-DETR), removing the 20 overlapping classes decreases performance at most by 1.5 pp, whereas removing 20 non-overlapping classes decreases less than 1 pp on the same benchmarks. This magnitude is, however, model dependent, GDINO drops by 4.1 pp on LostAndFound. This overall decrease in performance reflects the reduced diversity of

Table 7: Full vocabulary is the standard dataset with 15,604 outlier annotations. The 20 overlapping classes removal reduces the annotations to 12,863 and the non-overlapping ones to 12,832.

Model	**Annotation**	RO	RA	LF
SynOE-OD (DD:R50)	baseline — full vocabulary	48.7	26.0	32.0
	20 overlapping classes removed	47.2	25.8	30.5
	20 non-overlapping classes removed	48.3	25.9	31.5
SynOE-OD (GD:Swin-T)	baseline — full vocabulary	54.3	29.3	29.2
	20 overlapping classes removed	53.8	28.4	25.6
	20 non-overlapping classes removed	53.8	27.7	25.1

outliers in the training set rather than the absence of overlapping concepts. SynOE-OD therefore generalizes to unseen concepts and benefits from diversity in the outlier vocabulary.

Proportion of synthetic outliers. The proportion of synthetic outliers determines the balance between ID and OOD performance, which are optimal at different levels. How many outliers are needed is a key parameter of SynOE-OD, and we measure how much ID performance is exchanged for them. We vary the fraction of the training data that carries synthetic outliers while keeping the image count fixed at $10{,}702$ (table 8). A quarter of the outliers already delivers most of the OOD performance, with ID mAP essentially unchanged. RoadAnomaly does not improve beyond this point, whereas RoadObstacle reaches its plateau at half the outliers, and LostAndFound continues to improve up to the full set. ID performance decreases monotonically by up to 2 pp. Most of the OOD capability is achieved by using only a quarter of the outliers at no measurable ID cost. Preserving ID performance to maximum favors a smaller proportion of outliers, whereas maximizing OOD performance requires a larger proportion of outliers.

Table 8: Effect of synthetic outliers proportion. The outlier percentage shows the inpainted images retained. This corresponds to 3.3%, 6.5%, 9.2% and 13.2% of the annotations being OOD and every other image is the original scene with its ID annotations.

Outliers	RO	RA	LF	NU
0 % (ID-only)	12.0	2.4	5.2	40.4
25 %	46.4	26.1	29.9	40.0
50 %	48.2	26.2	30.4	39.9
70 %	48.1	25.5	31.0	39.2
100 %	48.7	26.0	32.2	38.7

Annotation refinement. The annotation refinement step in our method is crucial and affects both ID and the OOD performance of SynOE-OD, see appendix table 20. Without bounding box refinement, OOD AP drops significantly while ID performance stays almost unchanged $33.7\,\%$ against $34.5\,\%$. Label reassignment identifies 1,307 samples for *car*, that are regenerated as a *car*, and restoring them increases *car* AP from $51.0\,\%$ to $54.7\,\%$. Without restoring these regions, the generated outliers contain modified ID objects and OOD class is contaminated with ID objects. Therefore, annotation refinement is essential to maintain a clear separation between ID and OOD supervision and to prevent training ID objects as OOD.

5 Conclusion

We address out-of-distribution object detection in street scenes and present SynOE-OD, which jointly detects ID and OOD objects with a single detector and requires no additional auxiliary branch at inference. Synthetic outliers are inpainted into street scenes with Stable Diffusion, their annotations are refined with an open-vocabulary detector, and the detector is fine-tuned with one additional class. Evaluations on the OoDIS benchmark show that our approach substantially surpasses existing zero-shot OVOD baselines, confirming that zero-shot detection alone is insufficient for OOD object detection, and that scoring based methods alone cannot recover missed OOD objects. These improved results hold consistently across closed-set and open-vocabulary detectors and training datasets, while the cost in ID performance remains small. Synthetic outliers prove effective mainly through accurate annotations and semantic diversity, and the resulting detection capability generalizes to real, unseen road obstacles.

AI USE STATEMENT

We used generative AI tools (Claude, Copilot) in this manuscript to find typos and grammatical mistakes, to generate LaTeX tables, to search for related work, to verify code, support in implementing experiments, to generate the outlier vocabulary section A, and help in shortening the text. We did not use them to write the paper, to propose the idea of this work, to generate the full coding pipeline, to design experiments, or to produce the core methodological or scientific claims of this work. All AI-assisted output was reviewed by the authors. Related work suggested was checked against the original publications. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Implementation details for the generation pipeline and for fine-tuning each detector are given in section A, including crop sizes, prompt template, learning rates, batch sizes and schedules. The complete outlier vocabulary is given in section A. The OoDIS results are obtained through the benchmark's evaluation server and LostAndFound is evaluated locally under the same protocol.

REFERENCES

Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In *German conference on pattern recognition*, pp. 33–47. Springer, 2019.

Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Dense open-set recognition based on training with noisy negative images. *Image and vision computing*, 124:104490, 2022.

Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving. In *proceedings of the IEEE/CVF international conference on computer vision workshops*, pp. 0–0, 2019.

Daniel Bogdoll, Maximilian Nitsche, and J Marius Zöllner. Anomaly detection in autonomous driving: A survey. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4488–4499, 2022.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation. *arXiv preprint arXiv:2104.14812*, 2021a.

Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *Proceedings of the ieee/cvf international conference on computer vision*, pp. 5128–5137, 2021b.

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

Pau de Jorge, Riccardo Volpi, Puneet K Dokania, Philip HS Torr, and Grégory Rogez. Placing objects in context via inpainting for out-of-distribution segmentation. In *European Conference on Computer Vision*, pp. 456–473. Springer, 2024.

Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. 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.

Xuefeng Du, Gabriel Gozum, Yifei Ming, and Yixuan Li. Siren: Shaping representations for detecting out-of-distribution objects. *Advances in neural information processing systems*, 35:20434–20449, 2022a.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022b.

Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *Advances in Neural Information Processing Systems*, 36:60878–60901, 2023.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9235–9244, 2022.

Max Gutbrod, David Rauber, Danilo Weber Nunes, and Christoph Palm. Openmibood: Open medical imaging benchmarks for out-of-distribution detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 25874–25886, 2025.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Jens Henriksson, Stig Ursing, Murat Erdogan, Fredrik Warg, Anders Thorsén, Johan Jaxing, Ola Örsmark, and Mathias Örtenberg Toftås. Out-of-distribution detection as support for autonomous driving safety lifecycle. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*, pp. 233–242. Springer, 2023.

Sadia Ilyas, Ido Freeman, and Matthias Rottmann. On the potential of open-vocabulary models for object detection in unusual street scenes. In *European Conference on Computer Vision*, pp. 201–217. Springer, 2024.

Brian KS Isaac-Medina and Toby P Breckon. Dream-box: Object-wise outlier generation for out-of-distribution detection. *arXiv preprint arXiv:2504.18746*, 2025.

Xue Jiang, Feng Liu, Zhen Fang, Hong Chen, Tongliang Liu, Feng Zheng, and Bo Han. Negative label guided ood detection with pretrained vision-language models. *arXiv preprint arXiv:2403.20078*, 2024.

KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5830–5840, 2021.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Jinglun Li, Kaixun Jiang, Zhaoyu Chen, Bo Lin, Yao Tang, Weifeng Ge, and Wenqiang Zhang. Synthesizing near-boundary ood samples for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4496–4506, 2025.

Yi Li and Nuno Vasconcelos. Background data resampling for outlier-aware classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13218–13227, 2020.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Wenteng Liang, Feng Xue, Yihao Liu, Guofeng Zhong, and Anlong Ming. 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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Jiahui Liu, Xin Wen, Shizhen Zhao, Yingxian Chen, and Xiaojuan Qi. Can ood object detectors learn from foundation models? In *European Conference on Computer Vision*, pp. 213–231. Springer, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024b.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Xixi Liu and Christopher Zach. Tag: Text prompt augmentation for zero-shot out-of-distribution detection. In *European Conference on Computer Vision*, pp. 364–380. Springer, 2024.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Thibaut Loiseau, Tuan-Hung Vu, Mickael Chen, Patrick Pérez, and Matthieu Cord. Reliability in semantic segmentation: Can we use synthetic data? In *European Conference on Computer Vision*, pp. 442–459. Springer, 2024.

Hayk Manukyan, Andranik Sargsyan, Barsegh Atanyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Hd-painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. In *International Conference on Learning Representations*, volume 2025, pp. 96301–96330, 2025.

Alexey Nekrasov, Alexander Hermans, Lars Kuhnert, and Bastian Leibe. Ugains: Uncertainty guided anomaly instance segmentation. In *DAGM German Conference on Pattern Recognition*, pp. 50–66. Springer, 2023.

Alexey Nekrasov, Rui Zhou, Miriam Ackermann, Alexander Hermans, Bastian Leibe, and Matthias Rottmann. Oodis: Anomaly instance segmentation and detection benchmark. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2764–2771, 2025. doi: 10.1109/ICRA55743.2025.11128111.

Philipp Oberdiek, Matthias Rottmann, and Gernot A Fink. Detection and retrieval of out-of-distribution objects in semantic segmentation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, pp. 328–329, 2020.

Philipp Oberdiek, Gernot Fink, and Matthias Rottmann. Uqgan: A unified model for uncertainty quantification of deep classifiers trained via conditional gans. *Advances in Neural Information Processing Systems*, 35:21371–21385, 2022.

OpenAI. Chatgpt. `https://www.openai.com`, 2024. Large language model.

Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In *2020 international conference on systems, signals and image processing (IWSSIP)*, pp. 237–242. IEEE, 2020.

Peter Pinggera, Sebastian Ramos, Stefan Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. Lost and found: detecting small road hazards for self-driving vehicles. in 2016 ieee. In *RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1099–1106.

Shyam Nandan Rai, Fabio Cermelli, Dario Fontanel, Carlo Masone, and Barbara Caputo. Unmasking anomalies in road-scene segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4037–4046, 2023.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

RunwayML. Stable Diffusion Inpainting. `https://huggingface.co/runwayml/stable-diffusion-inpainting`, 2023. Accessed: 2025-05-16.

Youssef Shoeb, Azarm Nowzad, and Hanno Gottschalk. Out-of-distribution segmentation in autonomous driving: Problems and state of the art. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4310–4320, 2025.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International conference on machine learning*, pp. 20827–20840. PMLR, 2022.

Leitian Tao, Xuefeng Du, Xiaojin Zhu, and Yixuan Li. Non-parametric outlier synthesis. *arXiv preprint arXiv:2303.02966*, 2023.

Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*, 2019.

Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1802–1812, 2023a.

Qizhou Wang, Zhen Fang, Yonggang Zhang, Feng Liu, Yixuan Li, and Bo Han. Learning to augment distributions for out-of-distribution detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Samuel Wilson, Tobias Fischer, Feras Dayoub, Dimity Miller, and Niko Sünderhauf. Safe: Sensitivity-aware features for out-of-distribution object detection. In *Proceedings of the ieee/cvf international conference on computer vision*, pp. 23565–23576, 2023.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Geng Yu, Jianing Zhu, Jiangchao Yao, and Bo Han. Self-calibrated tuning of vision-language models for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 37:56322–56348, 2024.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

Jingyang Zhang, Nathan Inkawhich, Randolph Linderman, Yiran Chen, and Hai Li. Mixture outlier exposure: Towards out-of-distribution detection in fine-grained environments. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 5531–5540, 2023.

Wenjie Zhao, Jia Li, Xin Dong, Yu Xiang, and Yunhui Guo. Segment every out-of-distribution object. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3910–3920, 2024.

Orr Zohar, Kuan-Chieh Wang, and Serena Yeung. 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.

A IMPLEMENTATION DETAILS

Stable Diffusion. The inpainting quality is greatly affected by the size of the crop region. We use three different sizes, i.e., for small objects 128×128 pixels, while for medium sized 256×256 pixels and for large objects 512×512 pixels. However, the Stable Diffusion model has a base resolution of 512×512 pixels; thus, the input to the model is always resized to 512×512 beforehand. The number of inpainted objects for each image is selected randomly and often times there is more than one inpainted OOD object. Before inpainting multiple objects in a single image, we make sure that there is a minimum distance between the cropped regions, such that the inpaintings are spatially separated. The minimum distance is based on the largest crop size, i.e., 512×512 pixels, that enforces a minimum center distance of 512 pixels. The resulting inpaintings are shown in fig. 4



Figure 4: Illustration of the inpainted synthetic outliers on NuImages. The first row shows the ID objects being replaced with OOD objects of different sizes while maintaining a minimum distance. The second row shows objects inpainted directly into the free-space i.e., the road region.

Object detectors. All detectors are fine-tuned on 3 GPUs with AdamW, a cosine-anealing learning schedule and an exponential moving average of the weights. For DINO-DETR with a ResNet-50 backbone, we kept the backbone stages frozen and the base learning-rate is set to 2×10^{-4} with a linear warm-up for the initial 250 iterations with a starting factor 0.001, the batch size is 6 per GPU, the EMA momentum is 5×10^{-3}. DINO-DETR with Swin-L and GDINO with Swin-T are fine-tuned with backbone stages unfrozen, a base learning rate of $1e-4$ and no warm-up. Swin-L uses a batch size of 2 per GPU, an EMA momentum 2×10^{-3} and GDINO uses a batch size of 6 per GPU with 5×10^{-3} EMA momentum. The unfrozen backbones and as well as the language model of GDINO are trained at $0.1\times$ the base learning rate. For DINO-DETR with ResNet-50, the sampling offsets of the deformable attention are also trained $0.1\times$ the base learning rate.

Datasets. The three ID datasets differ in acquisition: NuImages is derived from NuScenes with 2D labels, Cityscapes was captured in German cities in daylight and dry weather, and BDD100K was recorded in the USA across varied weather and lighting conditions. For NuImages we use the front camera images, which are consistent in camera view with the other two datasets. From BDD100K, we randomly selected 2K images, to prevent it from dominating the combined training set, and from Cityscapes 2,975 finely annotated training images. In analogy to the NuScenes benchmark, we use its six vehicle classes and the pedestrian class, which are *bicycle*, *bus*, *car*, *motorcycle*, *trailer*, *truck* and *pedestrian*. We exclude *cones*, *barriers* and *construction*, those are not typical ID street scene objects and are treated as OOD in RoadAnomaly.

Baselines. Table 9 groups the baselines by what each one controls for. The ID-only model has no OOD class, so every prediction is remapped to that category before.

Table 9: Baselines and what each accounts for.

Baseline	Comparison
ZS-OVOD	OoDIS
UGainS	OoDIS
VOS	training-based OOD
ID-only (ours)	effect of inpainted data

Fine-tuning Configuration. We vary the number of frozen backbone stages, the learning-rate schedule and the transformer dropout to find the setup whose OOD performance is most stable across epochs (table 11, table 10). The fully frozen ResNet50 backbone with cosine annealing is more stable across epochs, whereas for Swin-T and Swin-L unfrozen is better.

Table 10: Backbone freezing and learning-rate warm-up for DINO-DETR with Swin-L on inpainted NuImages. The standard configuration is the mean over three runs; each variant changes one setting.

Configuration	RO	RA	LF
Standard (no warmup, unfrozen)	0.578	0.264	0.306
+ LinearLR warmup	0.563	0.260	0.266
`frozen_stages=2`	0.575	0.241	0.275
`frozen_stages=1`	0.574	0.232	0.274

Table 11: Ablation grid - ID and OOD metrics averaged over seeds (24, 42). 'Frozen' refers to the backbone stages. The dropout is added to the transformer encoder, decoder. The model trained is DINO-DETR with ResNet50 on NuImages.

Config	Epoch	ID mAP	RO	RA	LF
A (frozen=1, flat LR)	1	32	47	26	30
	3	38	43	22	21
B (frozen=4, flat LR)	1	31	47	26	32
	3	36	44	23	23
C (frozen=4, cosine+warmup)	1	30	47	26	32
	3	36	46	24	25
D (frozen=4, cosine-only)	1	31	46	26	30
	3	37	46	23	23
E (frozen=4, dropout=0.1, warmup+cosine)	1	29	47	25	35
	3	35	47	24	27
F (frozen=4, cosine-only, drop=0.1)	1	31	47	25	33
	3	36	46	23	23

GDINO learning rate. The learning rate for GDINO trades OOD performance against ID perfromance. We show that a higher learning rate is better for ID performance while worse for OOD detection. See (table 12).

Table 12: Doubling the learning rate raises ID mAP and lowers OOD AP.

lr	Test/Val set	ep1	ep2	ep3	ep4
1e-4	RO	0.560	0.538	0.532	0.537
1e-4	RA	0.298	0.284	0.271	0.258
1e-4	LF	0.300	0.276	0.264	0.270
1e-4	ID mAP	0.432	0.460	0.472	0.477
2e-4	RO	0.550	0.532	0.526	0.524
2e-4	RA	0.298	0.269	0.259	0.246
2e-4	LF	0.280	0.248	0.249	0.257
2e-4	ID mAP	0.446	0.473	0.487	0.494

Exponential Moving Average. We keep a running average of the weights instead of the raw ones, which raises OOD performance and reduces its variation across seeds (table 13, table 14). At every training step, instead of keeping the raw weights, maintain an average of old weights that mixes the old weights and the new weights.

$$weights_{new} = 0.995 \times old_{average} + 0.005 \times current_{weights}$$

Table 13: Weight averaging against raw weights, DINO-DETR ResNet-50 on inpainted NuImages. Mean ± standard deviation over three seeds (242, 444, 2244) and $AP_{50..95}$ in percent.

	with EMA			**without EMA**		
Epoch	RO	RA	LF	RO	RA	LF
1	49.3±0.7	26.8±0.8	36.1±0.9	46.4±1.9	26.3±1.3	31.7±2.8
2	48.7±0.1	26.0±0.1	31.7±1.2	47.2±2.6	24.6±0.9	27.5±2.1
3	47.6±0.6	25.2±0.5	28.9±1.4	46.8±2.1	24.4±1.1	27.8±1.9
4	47.7±1.0	24.5±0.4	27.5±0.8	46.3±1.1	23.7±1.1	26.0±1.5

Table 14: ID validation mAP

Epoch	s444 no EMA	s444 EMA	s2244 no EMA	s2244 EMA
1	34.0	34.4	34.3	33.7
2	38.0	38.6	37.5	38.8
3	39.4	40.6	40.4	40.5
4	41.1	41.4	41.1	41.2

Seed Variance. We repeat training with different seeds n to establish how much of a difference between configurations can be attributed to run-to-run variation (table 15).

Table 15: Run-to-run variation at the reported epoch. Mean ± standard deviation over independently seeded runs, $AP_{50..95}$ in percent.

Model	**Training data**	n	RO	RA	LF
DD:R50	NuImages	3	48.7±0.11	26.0±0.08	31.7±1.15
GD:Swin-T	+ Cityscapes	2	55.6±0.57	28.2±1.34	33.5±0.14

Results on OoDIS. In table 16 we report the benchmark results by object size, showing which size range each benchmark is dominated by.

Table 16: OoDIS results, averaged over two seeds per model. S, M and L are the COCO area ranges.

	RoadObstacle			**RoadAnomaly**			**Fishyscapes**		
Method	S	M	L	S	M	L	S	M	L
SynOE-OD (DD:ResNet50)	38.7	69.6	76.3	3.7	13.6	60.9	18.3	59.3	52.7
SynOE-OD (DD:Swin-L)	**47.4**	**76.1**	**90.2**	2.5	10.9	61.2	19.0	58.2	59.4
SynOE-OD (GDINO:Swin-T)	46.1	74.7	87.0	**7.4**	**15.5**	**64.2**	**19.7**	**66.5**	**61.2**

Outlier vocabulary. The full outlier vocabulary in table 17: 90 object categories returned by ChatGPT for the query 'give a list of objects that are unusual for a street scene'.

Table 17: The full outlier vocabulary

alligator	fox	Pots pans
antelope	gerbil	Printer
armadillo	giraffe	pylon
badger	Glasses	pylon large
bear	goat	pylong small
bison	gorilla	rabbit
bloated plastic bag	guinea pig	raccoon
Bookshelf	hamster	rat
buffalo	hedgehog	rhinoceros
cardboard	helicopter	robot
cat	hippopotamus	sea lion
cavy	horse	seal
Chair	kangaroo	sheep
chicken	koala	Shoes
chimpanzee	Lamps	skateboard
Clock	lemur	sloth
Coffee table	lion	sloth bear
crate	manatee	Sofa
crate small	mole	squirrel
crocodile	monkey	styrofoam
Cutting board	monster	tamarin
deer	mouse	tapir
Desk	okapi	tiger
Dishes	opossum	tire
dog	orangutan	tortoise
elephant	ostrich	Vacuum cleaner
emu	panda	walrus
Fan	penguin	weasel
ferret	platypus	wolf
flamingo	porcupine	zebra

B ADDITIONAL ABLATIONS

Synthetic outliers vocabulary. Nine out of 20 overlapping classes are essentially road debris, i.e., pylons, cardboard, styrofoam, tires etc., and are the objects that RoadObstacle and LostAndFound contain, and eleven are animals which RoadAnomaly contains. The number of OOD annotations removed for the two setups is (2,741 and 2,772) respectively, and the image count is identical.

Unlabeled synthetic outliers. Synthetic outliers can end up in the images without an annotation if the annotation step queries GDINO on the full image, where it often misses those outliers. This ablation shows that outliers left unlabeled conflict with the learning of the OOD class, therefore justifying our design choice in section 3.3. We take the same 20 overlapping classes as above and, instead of discarding them, leave them unannotated in the images. RoadAnomaly falls from AP of 26.0% to 20.0% for DINO-DETR and from 29.3% to 20.1% for GDINO table 18. The reason is the mechanism described in the introduction, that unlabeled objects supervise against the OOD class and here measured for synthetic outliers.

Table 18: Effect of 2,741 unlabeled outliers. The parentheses show the difference from the baseline.

Model	RO	RA	LF
R50	0.483 (−0.4)	0.200 (−6.0)	0.300 (−2.0)
GD	0.514 (−2.9)	0.201 (−9.2)	0.268 (−2.4)

ID/OOD trade-off. In table 19 we report per-class ID performance alongside the annotation counts before and after inpainting. *Pedestrian* and *bus* keep all of their annotations and lose the least.

Table 19: Per class ID performance in $AP_{50:95}$. The *train* annotations are used for fine-tuning SynOE-OD and the *original* annotations for the ID baseline. The values in parentheses show the difference from the baseline.

Class	original	train	ID baseline	SynOE-OD
car	46,745	38,100	58.3	57.1 (−1.2)
pedestrian	32,210	32,210	42.3	41.4 (−0.9)
truck	8,029	5,386	42.8	40.6 (−2.2)
motorcycle	3,406	2,638	47.3	44.9 (−2.4)
bus	1,987	1,987	42.4	41.7 (−0.7)
bicycle	2,285	1,447	35.8	32.3 (−3.5)
trailer	636	429	16.6	13.4 (−3.2)
mAP	–	–	40.8	38.8 (−2.0)

Annotation refinement. We train on the unrefined inpainting boxes and without label reassignment to isolate what each of the two refinement steps contributes (table 20).

Table 20: Effect of bounding-box refinement (left) and label reassignment (right). Left: *Unrefined* refers to the original boxes and *refined* is the accurate boxes returned by GDINO. Right: *replaced* is the number of ID objects of each class that was inpainted over, *reassigned* is the number for which GDINO identified the regenerated object as the original class; the last two columns show the class AP without and with the label reassignment.

OOD boxes	**RO**	**RA**	**LF**	**ID**
unrefined	1.3	4.2	1.9	33.7
refined	46.9	24.4	29.3	34.5

Class	**replaced**	**reassigned**	**w/o**	**with**
car	9,952	1,307	51.0	**54.7**

C QUALITATIVE RESULTS.

In this section, we visualize the qualitative results. The detections in orange represent the OOD class, while the remaining predictions correspond to ID objects.



Figure 5: OOD and ID object detection results on the RoadAnomaly dataset.



Figure 6: OOD and ID object detection results on the RoadObstacle dataset.