Placing Objects in Context via Inpainting for Out-of-distribution Segmentation

Pau de Jorge[1], Riccardo Volpi[1], Puneet K. Dokania[2], Philip H.S. Torr[2], and Grégory Rogez[1]

[1] Naver Labs Europe https://europe.naverlabs.com/
[2] University of Oxford
pau.dejorge@naverlabs.com

Abstract. When deploying a semantic segmentation model into the real world, it will inevitably encounter semantic classes that were not seen during training. To ensure a safe deployment of such systems, it is crucial to accurately evaluate and improve their *anomaly segmentation* capabilities. However, acquiring and labelling semantic segmentation data is expensive and unanticipated conditions are long-tail and potentially hazardous. Indeed, existing anomaly segmentation datasets capture a limited number of anomalies, lack realism or have strong domain shifts. In this paper, we propose the Placing Objects in Context (POC) pipeline to realistically add *any* object into *any* image via diffusion models. POC can be used to easily extend any dataset with an arbitrary number of objects. In our experiments, we present different anomaly segmentation datasets based on POC-generated data and show that POC can improve the performance of recent state-of-the-art anomaly fine-tuning methods across several standardized benchmarks. POC is also effective for learning new classes. For example, we utilize it to augment Cityscapes samples by incorporating a subset of Pascal classes and demonstrate that models trained on such data achieve comparable performance to the Pascal-trained baseline. This corroborates the low synth2real gap of models trained on POC-generated images. Code: https://github.com/naver/poc

Keywords: Anomaly segmentation · OOD segmentation · Inpainting

1 Introduction

When we deploy autonomous agents such as robots or self-driving cars, we expose them to the unpredictable nature of the real world. Inevitably, they will encounter visual conditions that were not anticipated during training. In particular, the presence of unseen objects in the scene poses a significant safety hazard. For example, consider an unknown wild animal crossing the street and being classified by the model as "road". To tackle this issue, it is important to distinguish out-of-distribution (OOD) categories—*i.e.*, novel objects unseen during training—from the in-distribution (ID) ones. In the context of semantic image segmentation, this task is often referred to as *anomaly segmentation*.

Fig. 1: Samples from previous OOD datasets. FS Static has unrealistic OOD objects while RoadAnomaly and SMIYC datasets have strong domain shifts from Cityscapes. FS L&F (which manually inserts OOD objects) and StreetHazards (full simulation) have large set-up costs.

Dataset	No shift	OOD realism	Dynamic	Low cost
FS Static [3]	✓	✘	(✓)	(✓)
FS L&F [47]	(✓)	✓	✘	✘
SMIYC O. [5]	✘	✓	✘	✘
SMIYC A. [5]	✘	✓	✘	✘
RoadAn. [36]	✘	✓	✘	✘
StreetHaz. [21]	✘	(✓)	(✓)	✘
POC (ours)	✓	(✓)	✓	✓

Table 1: Comparison of anomaly test sets. We qualitatively compare datasets on four main axes. We score them as either good (✓), medium ((✓)) or bad (✘). Further discussion in Sec. 2.

Although several methods have been proposed that allow for segmenting anomalies [2, 12, 16, 27, 32, 35], accurately evaluating the performance of such methods is a challenge in itself. Given a model trained on a particular dataset, the aim is to test its ability to distinguish OOD categories in conditions that resemble the training domain. For example, to test a model trained for semantic segmentation of urban scenes, the ideal test set would be constituted by images showing OOD categories within an urban environment similar to the training one. Yet, potential anomalies follow a long-tailed distribution and it is inefficient, or even hazardous, to acquire and label images with arbitrary OOD objects.

Previous approaches to generating anomaly segmentation datasets can be grouped into three families: *Stitching and blending* OOD objects from other sources into images from the original dataset [3]; *Collecting images* from driving scenes and annotating OOD objects [5, 36, 47]; *Full simulation* of urban scenes with anomalies [21]. *Stitching and blending* is relatively inexpensive if OOD objects are segmented elsewhere, yet it often leads to unrealistic insertions (*e.g.*, object's size or illumination). *Collected images* contain real anomalies, but are expensive to acquire and have a significant distribution shift from the original dataset (it is not easy to find or replicate images following the original setup). *Full simulation* allows for perfectly blended objects but bears a high setup cost and results in severe drifts from the training distribution. See examples in Fig. 1.

Ideally, methods to generate anomaly segmentation datasets should satisfy four main desiderata: *i)* Minimal domain shift with respect to the training set—since large domain shifts may lead to underestimating anomaly segmentation capabilities; *ii)* Generating realistic images; *iii)* Allowing for a dynamic generation of new images with arbitrary OOD objects; *iv)* Incurring low setup costs.

This motivates us to introduce the **Placing Objects in Context** pipeline (POC), which enables practitioners to generate anomaly segmentation test sets by realistically inserting *any* object (OOD or ID) into *any* image on the fly. See a comparison of approaches followed to generate previous benchmarks and the proposed POC along our desiderata in Tab. 1.

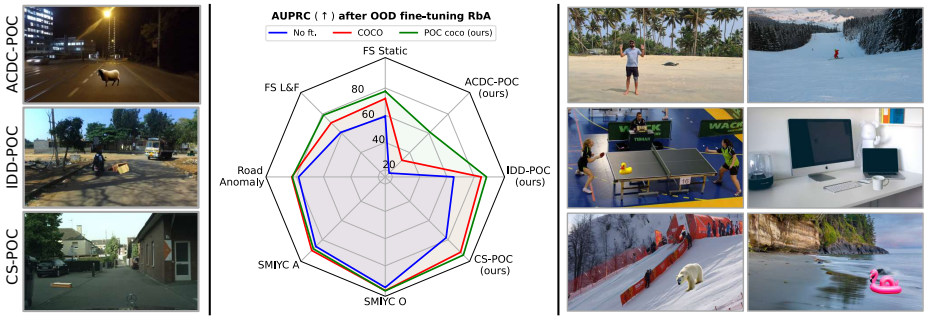

Fig. 2: Left: Samples of our POC-generated datasets. Top to bottom, inserted anomalies are "sheep", "dumped furniture" and "carton box". **Middle:** AUPRC on different anomaly segmentation datasets. We evaluate RbA [44] prior to fine-tuning, and after fine-tuning with COCO objects or POC-generated images. Fine-tuning with POC improves results on several benchmarks. **Right:** Beyond road scenes, POC can be applied seamlessly in diverse scenes. Clockwise, inserted objects are: "sea turtle", "person skiing", "white porcelain mug", "inflatable flamingo", "polar bear" and "rubber duck".

To realistically insert new objects, we control the location of the added object and apply only local changes to preserve the overall scene semantics. We utilize open-vocabulary segmentation [17] to select valid regions where the object can be placed, such as "the road". We then feed the selected region to an inpainting model [50] with a conditioning prompt, such as "a cat". After inpainting, we apply again the segmentation model to the modified area to automatically annotate the added object and detect generation failures, *i.e.*, when the object was not properly generated.

In our experiments, we show that fine-tuning on POC-generated data can significantly improve the performance of state-of-the-art anomaly segmentation methods—outperforming models fine-tuned via the standard practice of *stitching* COCO objects. We also present three POC-generated evaluation sets based on urban scene segmentation datasets [10, 54, 59], and benchmark different anomaly segmentation methods on them (see Fig. 2 (middle) for a first glimpse of results).

Finally, since POC can add arbitrary objects, we show it can be used to learn new classes. For instance, augmenting Cityscapes [10] images with animal classes leads to 93.14 mIoU on Pascal's test set [14] (using the same classes) *without seeing any real animal*, while directly training on Pascal yields 94.75—namely, models trained on POC-edited images exhibit a rather small synth2real gap.

2 Related work

We cover the relevant literature on methods and datasets for anomaly segmentation and on diffusion models—the research areas most related to our work.

Anomaly segmentation methods. Early works relied on approximating uncertainty via softmax probabilities [22, 33], model ensembles [30] or dropout

[15, 43]. Yet, models tend to be overconfident, resulting in high confidence also for OOD samples [18,25,45]. Alternative confidence measures have been proposed that either rely on logits [9, 21, 27, 38] or density estimators [31]. Another body of works reconstruct images with generative models and detect anomalies as discrepancies between original images and their reconstructions [19, 35, 36, 60]. Currently, the most promising methods use OOD data to fine-tune the models [6, 16, 58]. In particular, they crop OOD objects from COCO [34] and *stitch* them in Cityscapes images. In this work, we build on three state-of-the-art OOD fine-tuning methods [39, 44, 48] and combine them with POC.

Anomaly segmentation datasets. We compare existing anomaly segmentation datasets across four axes (*Domain shift*, *OOD realism*, *Dynamism* and *Set up cost*)—see Tab. 1 for a summary and Fig. 1 for sample images. The **Fishyscapes Static** [3] approach involves randomly stitching OOD objects from Pascal [14] onto Cityscapes images. While this method avoids domain shift, the stitched OOD objects lack realism. If OOD object images and masks are available, datasets can be generated *dynamically*. However, if new objects are required, they must be obtained from additional datasets or from the web incurring moderate *setup costs*. At the other end of the spectrum, **RoadAnomaly** [36] and **Segment-me-if-you-can (SMIYC)** datasets [5] contain real images with anomalies downloaded from the web. While this ensures *OOD realism*, it often leads to a large *domain shift*. Moreover, manual labelling of OOD objects has a significant *setup cost* and is *not dynamic*, *i.e.*, new images would need to be acquired and labelled to generate new samples. **Lost & Found** [47] mimics the Cityscapes setup to reduce domain shift, but OOD objects have been inserted artificially, leading to low variability and difficulty in scaling. **Street Hazards** [21] shows a fully simulated dataset that allows for *dynamic* generation of images while the simulation engine inserts OOD objects *realistically* in terms of lighting. Yet, pose and size of the object are pseudo-random, which is not always realistic and leads to a strong *domain shift* from simulation to real images. Moreover, it requires an accurate 3D model of all objects, which bears a significant *setup cost* and hinders scalability.

In contrast to prior art, our proposed pipeline is plug-and-play and allows for adding objects into images with *no setup costs*. Built on top of open-vocabulary models, it can *dynamically* insert any object by changing the text prompts. We observe both qualitatively and quantitatively that our pipeline leads to greater *OOD realism* and applying inpainting also helps to mitigate *domain shift*.

Diffusion models. Introduced by Sohl-Dickstein *et al.* [55], diffusion models have led to unprecedented quality in image generation [11, 23, 52, 56]. In particular, text-to-image models condition the image generation or image editing process on a given text prompt [46, 49, 50, 53]. Image inpainting methods, which insert objects locally by only modifying masked regions of an image [1, 49, 50], are particularly relevant to our goal. General-purpose editing models that can edit images based on text prompts [4, 42] are also related to our work. Among

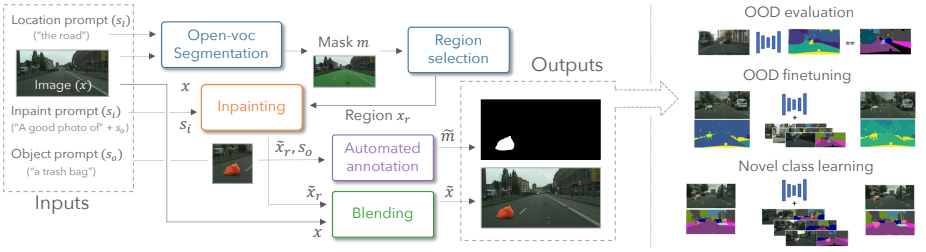

Fig. 3: Illustratation of our POC pipeline and applications. Our pipeline builds on top of inpainting and open-vocabulary segmentation models to insert arbitrary objects into images realistically. The resulting images can be used for different tasks.

the latter, InstructPix2Pix [4] has recently shown very realistic results following text instructions, *e.g.*,"make the photo look like it was taken at sunset". Yet, we found that it often fails to add new objects to the scene *e.g.*,"add a dog on the street" (see more details and illustrative images in Appendix C). In our work, we build on Stable Diffusion [50], demonstrating its ability to realistically insert objects into an image without requiring further training *when combined with other components* (our POC pipeline).

Other works have explored the usage of Stable Diffusion [50] to handle OOD classes. Similar to us, Du *et al.* [13] use text-to-image models to generate OOD images for classification and Karazija *et al.* [28] for zero-shot semantic segmentation. While the latter [28] relies on a frozen feature extractor and focuses on zero-shot segmentation, we *extend* an existing dataset with new classes. Different from the former [13], we target OOD in segmentation and rather than generating fully OOD images, we insert OOD objects into images realistically. Concurrent to our work, Loiseau *et al.* [41] also propose to leverage generative models to evaluate the reliability of semantic segmentation models, while their work focuses more broadly on evaluating different aspects of uncertainty estimation, we focus on anomaly segmentation and dataset extension.

3 Placing Objects in Context (POC)

We now present our proposed pipeline, Placing Objects in Context (POC). To recap, our desiderata to generate OOD datasets are *i)* minimal shift with respect to the training set, *ii)* realism, *iii)* dynamic generation and *iv)* low set-up costs.

3.1 The POC pipeline

Our pipeline builds on top of two open-source models with permissive licenses (an important detail towards open research): an inpainting model from Stable Diffusion (SD [50]) and an open-vocabulary segmentation model (GSAM) [17] based on SAM [29] and GroundingDINO [37]. In the following, we detail the main stages of our pipeline. See Fig. 3 for a comprehensive overview.

Selecting a region for inpainting. To realistically insert objects, it's crucial to find suitable locations for them (*i.e.*, a cat should not appear to be levitating). We use GSAM [17] to segment a suitable area for inserting the object based on a location prompt s_l, *e.g.*,"the road". Within this valid area, we select a region r with random size and location based on user-specified limits. To assess the gain in realism obtained by guiding the object location *vs.* picking a random location, we conducted a human study where participants were asked to choose between image pairs with/without guided location. We find that 43% of times the guided location was preferred *vs.* 18% for the random location; 39% of cases the preference was unclear—which is reasonable, since the road category tends to occupy a large percentage of the image. Additionally, note that generated objects differ when inpainting in different locations, which makes the comparison harder. We refer to Appendix D for more details and illustrative images.

Object inpainting. After selecting r, we crop a square around it (x_r) and apply SD to obtain $\tilde{x}_r$. The strong vision grounding from SD allows adding objects more realistically. For instance, we observe that the inpainting model tends to adjust the size of the object (*e.g.*, a bird will be much smaller than a garbage bin for the same region) although there is a tendency to fill all the inpainting area. We also observe that the illumination of the added object is adapted to the image. This is particularly noticeable in night images from ACDC (see Fig. 2).

Automated annotation. In most downstream applications, we need the corresponding mask of the added object for training or evaluation. Again, we rely on open-vocabulary segmentation to obtain $\tilde{m}$ based on an object prompt s_o. We observed that applying GSAM to the full image often leads to false positives; we obtain better results by applying it only to the inpainted region. Note that, by relying on open-vocabulary segmentation, we can provide more accurate labels than simpler methods like foreground/background segmentation [28] that cannot distinguish between a bicycle and its rider. During this step, we also reject images with generation failures (*e.g.*, the object was not generated or it was very unrealistic) if GSAM does not detect the generated object.

Object blending. While SD tends to preserve the details of the original image, it introduces slight modifications to the textures and sometimes noticeable changes, *e.g.*, in lane markings. To reduce undesired edits, we blend the original image x and the inpainted one $\tilde{x}$ as: $\tilde{x} := (1-m) \odot x + m \odot \tilde{x}$ where $:=$ indicates an update operation, $\odot$ is the element-wise product and $m = \mathcal{G}(\tilde{m})$ is the object mask convolved with a Gaussian kernel. This allows for a smooth transition between the inpainted object and the rest of the image while preserving some realistic local modifications, like shadows or reflections. We also considered applying an image2image (I2I) [24] generative model with an empty prompt after inpainting, but a human study showed that in 71% of the cases I2I blending did not improve results significantly, in 25% it introduced artifacts that significantly reduced realism and only in 4% of the cases participants preferred I2I blending (see Appendix E). In light of this, we only used the gaussian blending.

3.2 Generating datasets with POC

Although our pipeline can be used for multiple applications, we consider two main ones: *i)* extending datasets for *anomaly segmentation* and *ii)* learning new classes. In both cases, we use Cityscapes classes as the known one; regardless of the number of added classes, all POC-datasets have 3× the size of Cityscapes—*i.e.*, we augment each image three times with randomly sampled prompts. Note that this does not lead to any imbalance in terms of fine-tuning steps, since all training schedules have the same total amount of iterations.

Generation prompts. In all datasets we follow a similar approach. Each object class has an object prompt s_o, or several for diversity (*e.g.*,"car", "suv", "van" all belong to the Cityscapes class "car"). Then, we build the inpainting prompt as s_i = "A good photo of $\{s_o\}$". Given that the datasets we use are all for autonomous driving applications, we use the location prompt s_l = "the road" for all objects except the class "bird", which has unconstrained location. We found this simple approach to yield good results without further prompt engineering.

Generating OOD *test* sets. To evaluate anomaly segmentation methods, we generate three new test sets, namely *CS-POC*, *IDD-POC* and *ACDC-POC*. We start from the Cityscapes [10], IDD [59], and ACDC [54] test sets (which have increasing domain shift w.r.t. Cityscapes [26]) and use the same list of OOD objects to augment the images within the test set. Our motivation is to disentangle the difficulty in detecting OOD objects *vs.* the difficulty caused by a large distribution shift on the ID classes (while the OOD classes remain the same). Another goal is to showcase that POC can be used with different datasets seamlessly. All POC datasets contain 25 different OOD classes arbitrarily chosen to be plausible anomalies in an urban environment (*e.g.*, wild animals, garbage bags and bins, *etc.*). Given our OOD objects are all *synthetic*, following [3] we also add ID objects (*e.g.*, cars or persons) to ensure the model is not "simply" performing synth *vs.* real discrimination. See the full list in Appendix A.

Note that, while we focus on urban segmentation datasets for consistency with previous work, POC can be applied to arbitrary datasets (see Fig. 2, right)

Generating OOD *fine-tuning* sets. Recent OOD fine-tuning methods [39, 44, 48] use COCO classes not present in Cityscapes to extract OOD objects. For consistency, we generate *POC coco* using the names of COCO classes used in previous works as prompts to inpaint them with POC (as opposed to cropping and stitching from COCO images). Moreover, we generate *POC alt.*, an alternative fine-tuning dataset with different OOD classes (the ones used in our evaluation sets) to assess the dependence of the fine-tuning methods on the OOD objects.

Extending datasets to *learn new classes.* Given a dataset $\mathcal{D}$ with a set of classes $\mathcal{K}$, we consider the task of generating an *extended* dataset $\tilde{\mathcal{D}}$ with a set of classes $\tilde{\mathcal{K}} = \mathcal{K} \cup \mathcal{U}$. This dataset can be used to train models that may perform well on both $\mathcal{D}$ and on samples containing the additional classes from $\mathcal{U}$. Following the autonomous driving use-case, we extend the Cityscapes dataset with the 6 animal classes present in the PASCAL dataset (which we use for

evaluation). Our motivation is that wild animals cause accidents [51] and being able to segment them individually (not just as anomalies) might help preventing collisions. We call this dataset *POC-A*. We further add the same classes on the CS test set which we refer to as *CS extended*. Additionally, we generate POC-CS+A, where we inpaint both animal classes and Cityscapes classes.

FT	OOD data	FS Static			FS Lost&Found			SMIYC Anomaly			SMIYC Obstacle		
		F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓
M2A [48]	No ft.	65.4	60.1	7.4	45.5	36.8	13.9	88.6	93.0	3.9	69.3	73.6	6.9
	COCO	**82.7**	**88.4**	2.2	70.9	64.5	13.1	90.5	**94.7**	3.3	90.0	94.8	0.4
	POC alt.	82.6	87.4	3.1	74.5	68.8	11.4	**92.2**	93.8	**2.1**	90.8	95.3	0.3
	POC c.	82.0	87.0	**2.1**	**76.5**	**73.0**	**9.2**	88.8	92.1	8.4	**91.4**	**96.0**	**0.1**
RPL [39]	No ft.	21.0	13.9	38.5	5.4	1.6	66.3	50.2	53.0	39.6	46.0	45.2	3.2
	COCO	83.8	89.6	1.2	58.0	58.4	3.2	**73.4**	**78.3**	**18.1**	89.7	**94.3**	**0.3**
	POC alt.	88.4	94.4	0.7	**70.3**	**74.2**	2.7	69.4	78.2	26.9	**89.8**	93.6	0.9
	POC c.	**88.5**	**95.1**	**0.5**	69.5	69.2	**1.6**	62.7	68.0	46.8	87.8	92.3	0.9
RbA [44]	No ft.	58.2	59.2	17.7	63.7	61.0	10.6	**86.5**	86.9	86.4	92.8	95.9	0.2
	COCO	67.0	72.2	4.0	69.7	70.9	8.7	84.7	**91.1**	5.4	**94.9**	98.2	**.04**
	POC alt.	68.3	77.2	3.4	74.0	76.7	5.4	**86.5**	90.5	**4.4**	**94.9**	**98.4**	**.04**
	POC c.	**68.6**	**77.6**	**3.3**	**76.5**	**78.9**	**3.3**	85.2	89.4	5.0	94.6	**98.4**	**.04**

FT	OOD data	RoadAnomaly			Cityscapes–POC			IDD–POC			ACDC–POC		
		F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓	F1↑	AuPRC↑	FPR↓
M2A [48]	No ft.	54.8	55.7	52.2	40.9	35.4	13.1	45.6	42.9	12.0	18.6	10.3	26.7
	COCO	77.2	78.9	**18.5**	88.3	93.9	0.5	81.0	85.5	**1.0**	70.8	72.8	**6.2**
	POC alt.	**81.5**	**82.3**	36.7	**91.4**	**95.8**	**0.4**	82.8	87.7	1.1	**74.0**	**74.5**	7.6
	POC c.	78.9	78.0	24.6	89.1	94.7	**0.4**	**83.1**	**89.1**	1.3	72.8	72.0	8.4
RPL [39]	No ft.	24.4	15.0	70.4	22.7	13.8	44.9	8.0	3.5	61.4	3.3	1.3	79.0
	COCO	60.1	59.2	27.1	82.1	88.4	0.7	72.4	78.7	1.8	63.6	65.4	2.6
	POC alt.	**66.4**	**69.4**	21.7	86.1	93.2	0.5	79.7	85.8	1.0	**79.4**	**85.7**	**0.8**
	POC c.	61.4	64.2	**21.4**	**87.6**	**94.3**	**0.4**	**82.8**	**90.0**	**0.7**	76.0	81.2	1.3
RbA [44]	No ft.	72.8	78.4	11.8	73.5	77.9	3.7	65.5	65.1	78.9	24.2	18.7	90.0
	COCO	78.5	83.4	**8.3**	87.2	92.9	0.5	79.2	85.3	1.2	33.4	32.0	11.2
	POC alt.	**78.3**	**84.1**	**8.3**	89.3	95.0	0.4	83.7	**89.1**	**0.7**	55.0	58.4	**8.4**
	POC c.	77.3	83.0	8.8	**90.5**	**95.8**	**0.3**	**83.8**	**89.1**	0.9	**57.6**	**61.1**	9.1

Table 2: Anomaly segmentation results after OOD finetuning. We use three recent OOD fine-tuning methods (FT) and report results prior to fine-tuning (*No ft.*), after fine-tuning with *COCO* objects and using our POC pipeline to inpaint *COCO* objects (*POC c.*) or an alternative set of 25 objects likely to be found on the street (*POC alt.*). Best and second best numbers for each method are highlighted in **bold** and underlined, respectively. The best number over all methods is shaded in **gray**. Our pipeline improves performance in most cases. Moreover, *COCO* fine-tuning leads to significant improvements in our POC eval sets, consistent with previous benchmarks.

4 POC for anomaly segmentation

As already discussed, to reliably evaluate the risk of deploying a model in a certain scenario, we need to test on images with minimal distribution shift w.r.t. the original distribution and with realistic OOD objects. Similarly, we hypothesize that fine-tuning anomaly segmentation models with more realistic anomalies will improve results. To test this hypothesis, we take three recent methods for OOD fine-tuning that rely on stitching COCO objects into CS images; instead of stitching, we use our POC pipeline to generate the anomalies for fine-tuning.

4.1 Experimental setting

Anomaly segmentation methods. *RPL* [39] learns a module to detect anomalies via contrastive learning, on top of a segmentation network that is kept frozen. This allows improving OOD detection with minimal degradation to the closed-set performance. *Mask2Anomaly* (M2A) [48] and *RbA* [44] are both based on the novel Mask2Former architecture [8] which performs segmentation at the mask level, *i.e.*, by grouping pixels into "masks" and classifying the whole masks into the closed-set categories. This significantly reduces the pixel-level noise on the anomaly scores. RbA [44] performs OOD fine-tuning with a squared hinge loss while M2A [48] also uses contrastive learning. We use the original code with default settings for each method and only modify the fine-tuning dataset.

OOD fine-tuning data. For each method, we consider different ways of generating the anomaly fine-tuning datasets. First, we consider a baseline where no fine-tuning occurs (*No ft.* in our tables). Then, we consider fine-tuning on *COCO* stitching [6]. When we fine-tune using our POC-generate images, we consider two cases: *POC coco* (inpainting the same classes as *COCO* stitching) and *POC alt.* (inpainting alternative classes, more likely to be found on the street).

Anomaly datasets and metrics. For evaluation, we employ five commonly used datasets, namely Fishyscapes [3], Segment Me If You Can (object and anomaly) [5], Road Anomaly [36] and Lost and Found [47] already discussed in Sec. 2. Additionally, we evaluate on our POC-generated test sets. Following previous work [6, 39], we compute three different metrics: maximum F1 score over all thresholds ($F1^*$), Area under the Precision-Recall Curve (AuPRC) and False Positive Rate at 95% recall (FPR).

4.2 Results

POC improves OOD detection. In Tab. 2, we show that fine-tuning with *POC coco* is remarkably better than the *No ft.* baseline despite only using synthetic anomalies. Moreover, *POC coco* brings important improvements over COCO fine-tuning in some settings. For instance, in FS Static (for RPL or RbA), in SMIYC Obstacle (for M2A or RbA) and in FS Lost & Found (for all methods). In other settings, it is competitive with COCO fine-tuning—except in SMIYC Anomaly (for RPL), where we observe a significant drop. One reason may be the strong domain shift in SMIYC Anomaly, limiting the benefits of using more realistic data. On the other hand, in FS Lost & Found, which has the closest setting to Cityscapes and real OOD objects, we carry the largest improvements.

Robustness to the choice of OOD classes. We observe that fine-tuning with *POC alt.*, with different OOD classes than COCO, leads to strong results, improving over the COCO baseline in several settings and sometimes surpassing *POC coco*. This shows that these methods are somewhat robust to the choice of the OOD classes. The flexibility of the POC pipeline allows studying which classes are best depending on the use-case, a possible direction for future work.

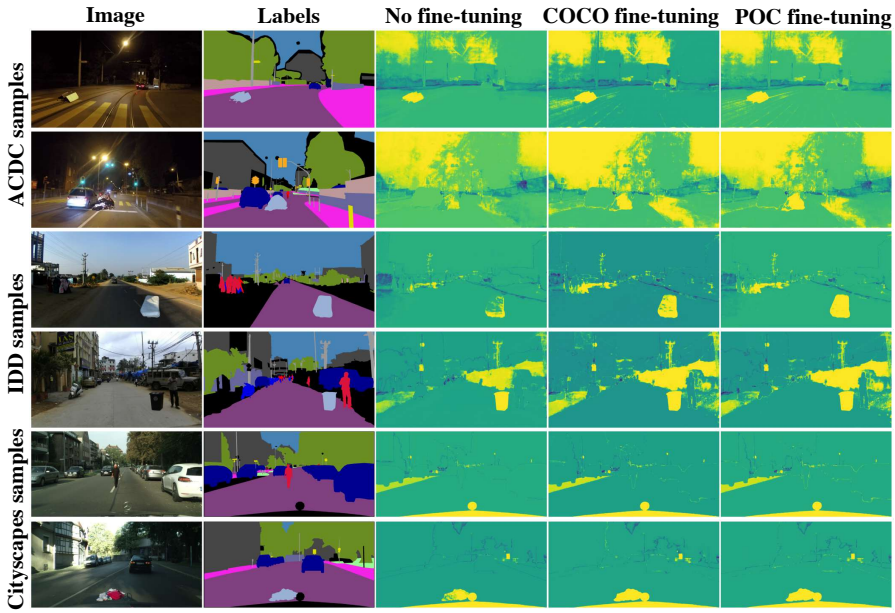

Fig. 4: Anomaly score maps. Per-pixel anomaly scores on POC-generated images obtained with M2A [48], before and after fine-tuning with COCO and POC data. COCO and POC fine-tuning have notable improvements over the *No ft.* baseline, *e.g.*, note the garbage bag or matress in second and third images.

Finally, note that the best score of all methods (highlighted with gray background) corresponds to one of the POC fine-tuning in most settings.

Performance on POC test sets. As expected, POC fine-tuning performs best on our POC-generated test sets (see Cityscapes-POC, ACDC-POC and IDD-POC in Tab. 2). Yet, fine-tuning with COCO also leads to notable improvements, similar to the ones observed in previous datasets. This suggests that POC datasets accurately reflect anomaly segmentation capabilities and can be used to efficiently build test sets. Interestingly, we also observe that *POC alt.* does not always outperform *POC coco* despite sharing the same OOD classes as the POC test sets. One explanation could be that *POC coco* has more OOD classes (80 compared to 25 in *POC alt.*), hence, more diversity.

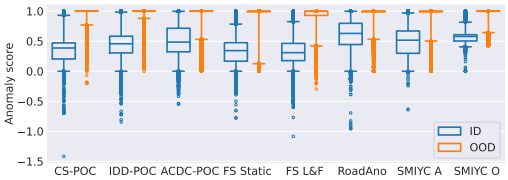

Fig. 5: Boxplots of anomaly scores. All datasets have consistently very high scores for OOD pixels while ID pixels of datasets with strong distribution shifts also have shifted scores. Thus, distribution shifts may lead to underestimated performance.

Domain shifts hinder anomaly segmentation. Considering our synthetic evaluation datasets, moving from POC-CS to POC-IDD and POC-ACDC we observe a drop in performance in all methods. Since the sets contain the same anomaly classes, the drop is due to the domain shift between train and evaluation rather than hard-to-detect anomalies. Additionally, in Fig. 5 we show boxplots of the predicted anomaly scores (higher numbers reflecting higher chance of anomaly) for ID *vs.* OOD pixels. OOD pixels have very high scores independently of the dataset, while ID scores vary significantly between datasets. Datasets with strong domain shifts (*i.e.*, SMIYC Anomaly, RoadAnomaly and POC-ACDC) carry larger ID anomaly scores. While anomaly segmentation under domain shift might also be an interesting task, we argue that, in practice, agents will be deployed in areas well represented by the training set. To accurately evaluate the risk represented by anomalies, there should be no domain shift.

Anomaly segmentation maps. In Fig. 4 we show per-pixel anomaly scores on POC-generated images computed with Mask2Anomaly [48] prior to fine-tuning and after fine-tuning either with COCO or POC data. Aside from the inpainted anomalies (labelled in gray), we observe that in ACDC (Top) the night sky has a particularly high score and in IDD (Middle) a peculiar instance of a known class (the all-road car) is also highlighted, potentially misleading anomaly segmentation. More examples can be found in Appendix H.

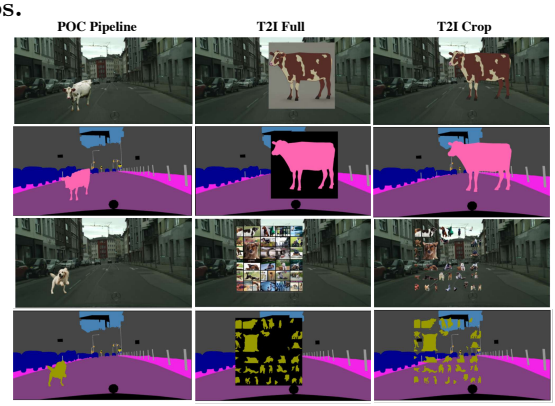

Fig. 6: Training image samples. Text2Image (T2I) compositions are often unrealistic. Moreover, sometimes they are misaligned with the caption, *e.g.*, "an image of a dog" leading to a *collage* of dog images. More samples in Appendix G.

5 POC to learn new classes

Besides anomaly segmentation, another natural application of our POC pipeline is dataset *extension*. In this final experimental section, we study this task in two dimensions: *i)* adding novel objects to learn new classes and *ii)* adding instances of existing classes to improve generalization.

5.1 Experimental settings

***Extended* datasets.** Like in our previous experiments, we use Cityscapes as our base dataset. Motivated by autonomous driving, we are interested in learning new classes that could potentially cause road accidents if left undetected.

Thus, we add *animal* classes to obtain *POC A* and *POC CS+A* (animal and Cityscapes classes). Similar to OOD detection, finding a test set is challenging (*i.e.*, Cityscapes images with animals). Inspired by Karazija *et al.* [28], we use Pascal [14] animal classes to assess performance of POC-trained models. For completeness, we also evaluate on the test set of Cityscapes *extended* with POC *(CS ext.)*.

T2I baseline. Karazija *et al.* [28] use text2image diffusion models to cluster the features of a frozen model for zero-shot segmentation. As this work closely relates to our task, for completeness we include two augmented datasets in our experiments: *T2I Full* and *T2I Crop* where we take the Full (or cropped) image generated by a T2I model and stitch it into Cityscapes images (we use the text2image model from the same work used in our POC [50]). Fig. 6 shows a comparison of the synthetic datasets. T2I baselines often lead to unrealistic object size and position. Moreover, T2I models often generate images misaligned with our goal (*e.g.*, a composition of many small dog images). We generate labels with the same open-vocabulary model as our POC pipeline, which is more flexible than foreground/background segmentation—suggested in [28].

Arch.	Train set	CS	CS ext.	Pascal (A)	Pascal (CS)
DLV3+	Pascal(b)	–	–	80.6	85.8
	CS(b)	79.1	–	–	42.2
	T2IFull	77.1	73.2	28.3	23.5
	T2ICrop	78.2	81.5	26.2	25.1
	POCA	**80.0**	**84.1**	**30.4**	35.8
	POCCS+A	79.9	83.8	28.1	**53.6**
CNXT	Pascal(b)	–	–	94.4	93.9
	CS(b)	81.6	–	–	70.5
	T2IFull	82.3	83.0	60.4	65.6
	T2ICrop	82.4	86.0	61.1	67.6
	POCA	**82.9**	**86.7**	65.5	70.9
	POCCS+A	82.5	86.1	**69.8**	**82.4**
Segm.	Pascal(b)	–	–	94.8	91.4
	CS(b)	76.2	–	–	79.9
	T2IFull	77.2	79.5	82.0	74.3
	T2ICrop	77.6	81.3	76.0	75.4
	POCA	**78.5**	**82.3**	92.4	79.1
	POCCS+A	78.4	81.9	**93.1**	**89.6**
GSAM	(*)	42.0	41.1	75.1	76.1

Table 3: mIoU evaluation. We train three different models on Pascal and Cityscapes as baselines (b), and compare two Text2Image (T2I) generation methods with our POC. CS and A indicate Cityscapes and Pascal's animals classes, respectively. (*) We also compare with GSAM, the open vocabulary model used to automatically generate the masks of inpainted objects in POC, for completeness. GSAM is trained on multiple datasets [17].

Architectures. We perform experiments using three different architectures: *DLV3+* [7] with a ResNet101 backbone [20]; *ConvNext* [40], a recent convolutional architecture with UPerNet [61]; and *Segmenter* [57], a recent transformer architecture designed for segmentation. We use the default training settings for each model. Note that our goal is not to compare architectures, but rather to evaluate the various methods employed to expand the datasets.

5.2 Results

POC performs better than T2I baselines. In Tab. 3 we show the mIoU for each model after training using our different datasets, as well as two baselines trained on Cityscapes and Pascal. For all architectures, we find that training

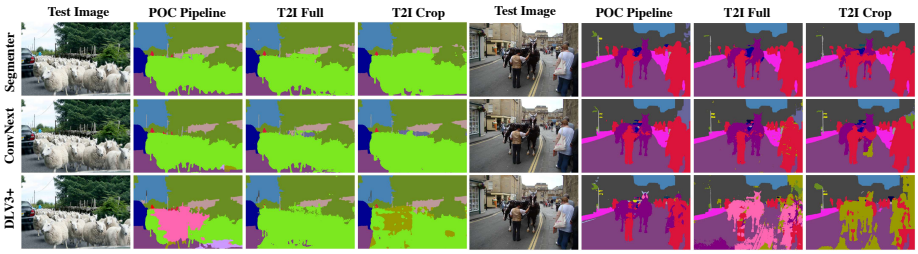

Fig. 7: Qualitative results. We present results on web images of road scenes with animals that have a milder domain shift from Cityscapes compared to Pascal images. Qualitatively, we observe less notable differences between CNXT and Segmenter but DLV3+ is still significantly worse. More results in Appendix I.

with POC datasets leads to better results than their T2I counterparts on CS extended and Pascal (A)—that is, Pascal's animal classes. We argue that this is due to the more realistic generated images (see Fig. 6). Interestingly, POC also improves performance on the original Cityscapes. Extending the set of classes might act as a form of regularization, but this requires further investigation.

Generalization is key to learn from synthetic data. We observe that the performance of DLV3+ trained on *POC A*—Cityscapes images with POC-inpainted animal classes—when evaluated on Pascal's animal classes (30.43 mIoU) is much lower than that of Segmenter (92.4 mIoU), which achieves an *mIoU competitive with the baseline trained directly on Pascal* (94.75). In order to understand that gap, we evaluate the performance of these models on the Pascal classes that are present on Cityscapes (*i.e.*, car, motorcycle, bike, person, train and bus). Interestingly, we observe that when trained on Cityscapes (without any inpainted classes) DLV3+ still has a much lower mIoU than Segmenter. This shows that DLV3+ has much less generalization capability independently of the synthetic classes. One could argue that perhaps the low performance of DLV3+ on Pascal is due to the large domain shift, however, in Fig. 7 we show qualitative results on web images of driving scenes, closer to the Cityscapes domain, and still observe that DLV3+ is notably worse than the other methods.

We hypothesize that, in order to learn transferable features from synthetic data, the generalization capability of the model plays a key role. Indeed, although generative models have improved the realism of generated content remarkably, there is still a synth2real gap. Thus, more robust models (*i.e.*, with strong transferability of learned features) may be able to extract more useful features rather than overfitting to brittle patterns in generated data.

We also note that the ResNet backbone in DLV3+ is pre-trained on the smaller Imagenet 1k, while ConvNeXt and Segmenter use Imagenet 21k; this might play a role, too. Nevertheless, there is also a gap between ConvNeXt and Segmenter in generalization, which cannot be explained by pre-training alone. Perhaps self-attention or image tokenization are relevant, but this would require a more in-depth analysis which is out of the scope of this work.

POC to augment existing classes. When evaluating the performance of Segmenter trained on *POC A* (containing the original Cityscapes classes extended with POC-added animal classes) we see a gap between Pascal (A) and Pascal (CS). This compels us to also inpaint Cityscapes classes with POC, obtaining *POC CS+A*. Training with the latter dataset, leads to remarkable improvements in Pascal (CS classes) in all three networks, significantly outperforming the baseline that was only trained on Cityscapes. Coupled with the milder but consistent improvement on the original Cityscapes, this indicates that our pipeline could also be helpful in improving in-distribution generalization.

Open vocabulary baseline. Since we rely on a open-vocabulary segmentation method (GSAM [17]) to label objects inserted with POC, we also evaluate its performance as a baseline. Interestingly, although we find the performance on Pascal reasonable, GSAM significantly underperforms other models on Cityscapes. We hypothesize this may be due to the one-vs-all nature of open-vocabulary predictors (where each class is predicted individually with a prompt) that does not perform well on complex images with many classes. Interestingly, the Segmenter model trained with POC significantly outperforms GSAM on Pascal.

Our dataset extension can be interpreted as a form of knowledge distillation from a teacher model (GSAM), which has been used to label the new classes in POC-extended datasets, while significantly improving its performance. On the other hand, note that the test setting is very different from the context in which we use GSAM in the POC pipeline, since in the latter we know the object that is being inpainted and only apply GSAM to the cropped region.

6 Concluding remarks

To accurately assess anomaly segmentation capabilities of models deployed in open-world settings, we argue that datasets should be realistic and carry a small domain shift with respect to the training distribution, as we show it can hinder OOD detection. Towards this goal, we introduce the Placing Object in Context (POC) pipeline, that allows adding *any* object into *any* image based on simple text prompting. POC uses diffusion models and open-vocabulary segmentation to achieve high realism and versatility.

We showcase POC's flexibility by generating three anomaly segmentation test sets: POC-CS, POC-IDD and POC-ACDC. Moreover, we observe that generating data for OOD fine-tuning with POC brings significant improvements in standard anomaly segmentation benchmarks. Beyond anomaly segmentation, we use POC-generated datasets to learn new classes without any real example. Interestingly, we observe that the combination of more realistic synthetic data with recent segmentation models with strong generalization capabilities can lead to a remarkable performance, competitive with training on real data.

In future work, we hope to better understand how to select the optimal set of anomalies for fine-tuning and how modern architectures can effectively rely on synthetic data to learn new classes.

Acknowledgements

This work is supported by the UKRI grant: Turing AI Fellowship EP / W002981 / 1. We would also like to thank the Royal Academy of Engineering.

References

1. Avrahami, O., Lischinski, D., Fried, O.: Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18208–18218 (2022)
2. Besnier, V., Bursuc, A., Picard, D., Briot, A.: Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15701–15710 (2021)
3. Blum, H., Sarlin, P.E., Nieto, J., Siegwart, R., Cadena, C.: 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)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18392–18402 (2023)
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. arXiv preprint arXiv:2104.14812 (2021)
6. Chan, R., Rottmann, M., Gottschalk, H.: 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 (2021)
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
8. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision (2022)
9. Corbière, C., Thome, N., Bar-Hen, A., Cord, M., Pérez, P.: Addressing failure prediction by learning model confidence. Advances in Neural Information Processing Systems **32** (2019)
10. 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. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
11. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
12. Di Biase, G., Blum, H., Siegwart, R., Cadena, C.: Pixel-wise anomaly detection in complex driving scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16918–16927 (2021)
13. Du, X., Sun, Y., Zhu, X., Li, Y.: Dream the impossible: Outlier imagination with diffusion models. arXiv preprint arXiv:2309.13415 (2023)

14. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (2012)
15. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059. PMLR (2016)
16. Grcić, M., Bevandić, P., Šegvić, S.: Densehybrid: Hybrid anomaly detection for dense open-set recognition. In: European Conference on Computer Vision. pp. 500–517. Springer (2022)
17. Grounded-SAM Contributors: Grounded-Segment-Anything. LICENSE Apache-2.0. `https://github.com/IDEA-Research/Grounded-Segment-Anything` (Apr 2023)
18. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International conference on machine learning. pp. 1321–1330. PMLR (2017)
19. Haldimann, D., Blum, H., Siegwart, R., Cadena, C.: This is not what i imagined: Error detection for semantic segmentation through visual dissimilarity. arXiv preprint arXiv:1909.00676 (2019)
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
21. Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., Song, D.: Scaling out-of-distribution detection for real-world settings. arXiv preprint arXiv:1911.11132 (2019)
22. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
24. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
25. Jiang, H., Kim, B., Guan, M., Gupta, M.: To trust or not to trust a classifier. Advances in neural information processing systems **31** (2018)
26. de Jorge, P., Volpi, R., Torr, P.H., Rogez, G.: Reliability in semantic segmentation: Are we on the right track? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7173–7182 (2023)
27. Jung, S., Lee, J., Gwak, D., Choi, S., Choo, J.: Standardized max logits: A simple yet effective approach for identifying unexpected road obstacles in urban-scene segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15425–15434 (2021)
28. Karazija, L., Laina, I., Vedaldi, A., Rupprecht, C.: Diffusion models for zero-shot open-vocabulary segmentation. arXiv preprint arXiv:2306.09316 (2023)
29. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
30. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems **30** (2017)

31. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems **31** (2018)
32. Liang, C., Wang, W., Miao, J., Yang, Y.: Gmmseg: Gaussian mixture based generative semantic segmentation models. Advances in Neural Information Processing Systems **35**, 31360–31375 (2022)
33. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
35. Lis, K., Honari, S., Fua, P., Salzmann, M.: Detecting road obstacles by erasing them. arXiv preprint arXiv:2012.13633 (2020)
36. Lis, K., Nakka, K., Fua, P., Salzmann, M.: Detecting the unexpected via image resynthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2152–2161 (2019)
37. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
38. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020)
39. 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 (2023)
40. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
41. Loiseau, T., Vu, T.H., Chen, M., Pérez, P., Cord, M.: Reliability in semantic segmentation: Can we use synthetic data? arXiv preprint arXiv:2312.09231 (2023)
42. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
43. Mukhoti, J., Gal, Y.: Evaluating bayesian deep learning methods for semantic segmentation. arXiv preprint arXiv:1811.12709 (2018)
44. Nayal, N., Yavuz, M., Henriques, J.F., Güney, F.: Rba: Segmenting unknown regions rejected by all. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 711–722 (2023)
45. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 427–436 (2015)
46. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
47. Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., Mester, R.: Lost and found: detecting small road hazards for self-driving vehicles. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1099–1106. IEEE (2016)

48. Rai, S.N., Cermelli, F., Fontanel, D., Masone, C., Caputo, B.: Unmasking anomalies in road-scene segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4037–4046 (2023)
49. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 **1**(2), 3 (2022)
50. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
51. Saad, W., Alsayyari, A.: Loose animal-vehicle accidents mitigation: Vision and challenges. In: 2019 International Conference on Innovative Trends in Computer Engineering (ITCE). pp. 359–364. IEEE (2019)
52. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH 2022 Conference Proceedings. pp. 1–10 (2022)
53. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. Advances in Neural Information Processing Systems **35**, 36479–36494 (2022)
54. Sakaridis, C., Dai, D., Van Gool, L.: Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10765–10775 (2021)
55. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International conference on machine learning. pp. 2256–2265. PMLR (2015)
56. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
57. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7262–7272 (2021)
58. 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. Springer (2022)
59. Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., Jawahar, C.: Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 1743–1751. IEEE (2019)
60. Xia, Y., Zhang, Y., Liu, F., Shen, W., Yuille, A.L.: Synthesize then compare: Detecting failures and anomalies for semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 145–161. Springer (2020)
61. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European conference on computer vision (ECCV). pp. 418–434 (2018)