# Finding Dino: A Plug-and-Play Framework for Zero-Shot Detection of Out-of-Distribution Objects Using Prototypes (Supplementary)

Poulami Sinhamahapatra, Franziska Schwaiger, Shirsha Bose, Huiyu Wang,
Karsten Roscher and Stephan Günnemann

## A. Details about creating OOD dataset via Inpainting

Due to absence of publicly available dataset in the rail ODD scene, we created an OOD dataset via inpainting OOD objects to the validation images of RailSem19 dataset [7]. Although generated images pertain to rail scene, but this method can be generically applied to any domain. The creation of the dataset involves relies on two methods - 'Inpaint-Anything' [6] and 'Segment Anything Model' (SAM) [4]. Inpaint-Anything takes some coordinates in an image and replaces the object which lies in the given coordinates. This object is replaced with the object that needs to be in-painted using text prompts, with the help of a diffusion model. Thus, the input image is changed to an image with the desired object given in the prompt at the specific location provided.

However, image generation using Inpaint-Anything is limited in the cases where there are no plausible objects to be replaced. Moreover, since the replaced object is inpainted, the corresponding OOD object mask is not obtained for utilising as the Ground-Truth (GT). To create the GT masks, we store the coordinate locations of the replaced object. Now, we leverage SAM by feeding the transformed image to it and also specifying the stored coordinate locations to generate the segmentation mask of the object associated with the coordinate locations. Thus, we create the image with the OOD object at the specified locations as well as the corresponding segmentation masks for the OOD objects in the image, as shown in Fig. 1.

## B. Additional Results

In this section, we show further results and insights of the comparative performance of our method zero-shot method PROWL and its variants resulting from combination unsupervised segmentation methods, STEGO and CutLER. In Sec. Firstly, we show the results for the segmentation outputs on the test set of In-Distribution (ID) datasets for each domain. Further, we show additional comparative results for OOD detection on the test set of the OOD datasets for Cityscapes [3].

### B.1. Performance comparison on the ODD classes of ID test datasets

Here, we show the prototype-based segmentation outputs for the ODD classes for different variants of PROWL on the test set of the ID datasets used for each domain, i.e. Cityscapes for road driving scene (Fig. 2) and RailSem19 for rail scene (Fig. 3).

Fig. 2 shows the segmentation from the test set images, for all 19 ODD classes of Cityscapes used to create the prototype feature bank. PROWL shows pixel-wise classification, whereas STEGO and CutLER provides semantic and instance segmentation masks combined with pixel-wise classification of PROWL. While both STEGO and CutLER generate unsupervised foreground masks, STEGO generates per-pixel output due to contrastive clustering of the ID train data whereas CutLER generates object boxes for foreground objects and then provides instance segmentation masks. Thus, CutLER provides segmentation for foreground object masks while ignoring background, like the *sky* is ignored in all the three test images as well as the *buildings* in last test image where they relatively lie in the background. Although, all these models have been trained for segmentation without labels, the overall segmentation is quite good. In PROWL, some noisy output is obtained (pixels in red shown in red) due to per-pixel classification based on ODD prototype classes. However, this is taken care of when combined with mask based evaluation using PROWL in STEGO and CutLER. *We note the importance of having good quality prototype feature bank as this reflects the performance in correctly classifying ODD classes or OOD pixels.* For example, in the segmentation GT for *road* in the prototype features include the test vehicle along with *Mercedes logo* and thus they have been labeled as *road*.

Fig. 3, we show segmentation outputs test split of RailSem19, for PROWL and PROWL with CutLER for the assumed simple ODD list with 6 classes - *train car, plat-*

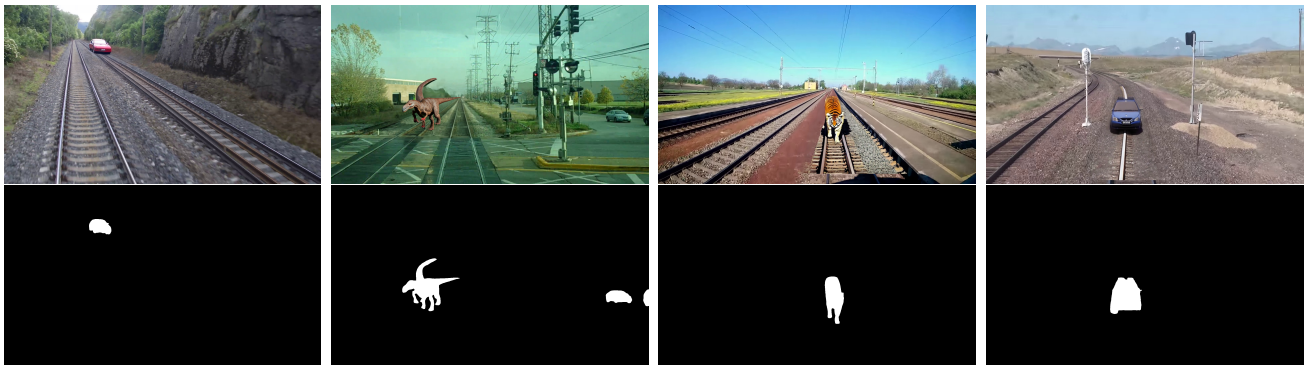

Figure 1. Sample images from OOD dataset created via inpainting OOD objects in RailSem19 dataset [7] (top-row) and the corresponding binary masks for the OOD objects (bottom-row).

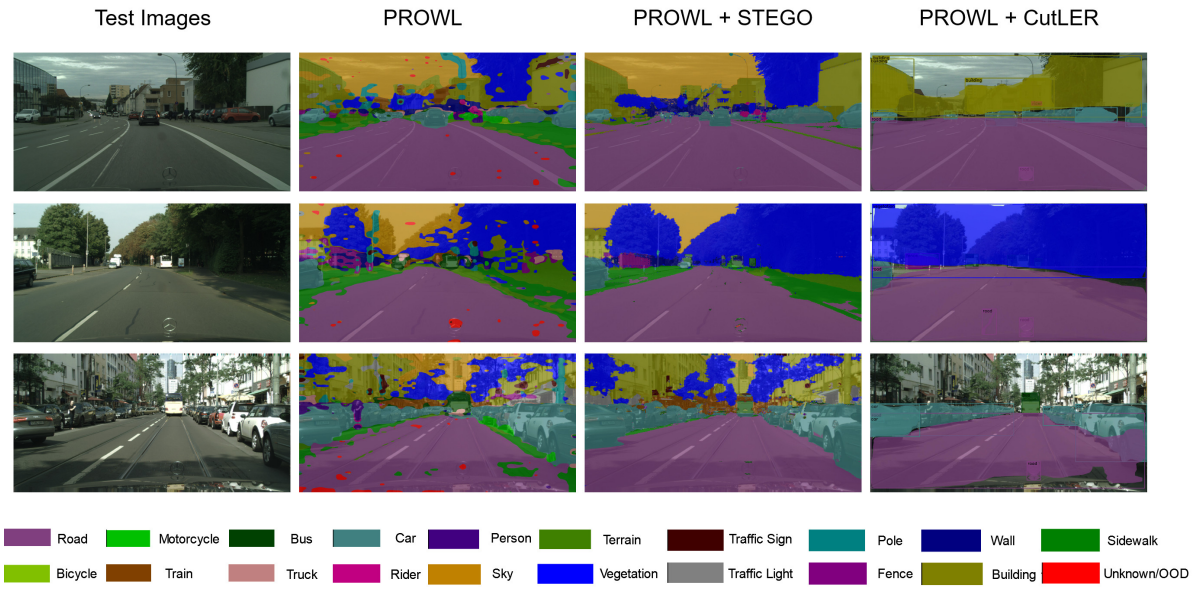


Figure 2. Performance comparison of our zero-shot methods on the segmentation outputs for the ODD classes on ID test dataset for road driving scene, i.e. Cityscapes [3].

*form, rail, fence, person, pole*. Since STEGO relies on unsupervised contrastive training on domain dataset and did not provide pre-trained model weights for RailSem19, we exclude it from comparison. We note although both zero-shot methods perform quite well on the ID test set, PROWL shows some OOD or unknown regions in red. This is primarily due to pixel-wise prototype matching where the pixels in red mostly correspond to classes like *vegetation*, not defined in our current ODD list. In PROWL + CUTLER, we do not directly detect *vegetation* as OOD as they are not detected as foreground masks and feature as background in the given test images since they are present over quite a distance. However, since *car* is not defined in ODD list, it is detected as OOD in the first test image. Thus, sufficiently defining ODD class list is crucial while detecting OOD / unknown objects to avoid false predictions.

## B.2. Performance comparison on OOD test datasets

Here, we show additional results for OOD detection using our PROWL and it's variants compared to supervised baseline (Maskomaly [1]) particularly for the *test* images for the given OOD datasets (RoadAnomaly and RoadObstacle) given in SMIYC benchmark [2], i.e Anomaly track and Obstacle track respectively, for Cityscapes [3] as ID dataset. We show only qualitative results on the test sets due to absence of GT in the benchmark. For fair comparison, we use fixed confidence threshold of 0.9 as suggested by authors of Maskomaly [1]. Similarly, for our methods - PROWL and its variants, we used inverse cosine similarity threshold fixed at 0.55. GT segmentation masks for OOD objects for these test images are not provided, thus we only show qualitative results in Fig. 4 and 5 with detected OOD objects in red.

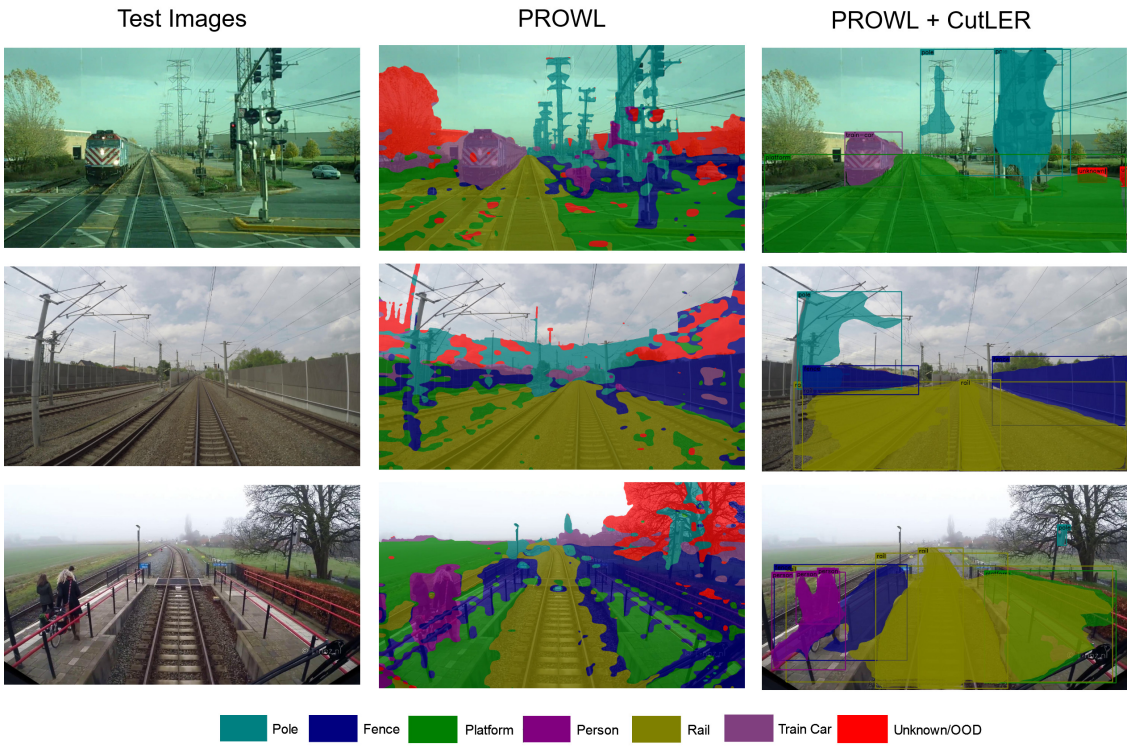


Figure 3. Performance comparison of our zero-shot methods on the segmentation outputs for the ODD classes on ID test dataset for rail scene, i.e. RailSem19 [7].

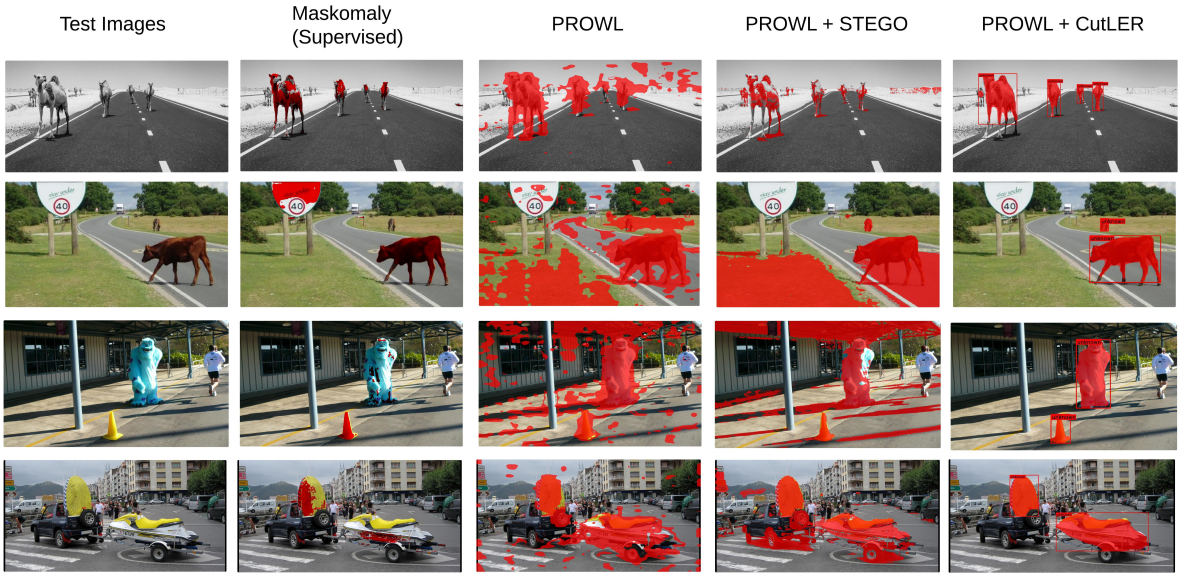


Figure 4. Performance comparison of our proposed zero-shot methods compared to supervised baseline for OOD object detection and segmentation on the test images of RoadAnomaly OOD (SMIYC-Anomaly Track) dataset. Detected OOD pixels are shown in red.

Fig. 4 show performance comparison on the RoadAnomaly dataset where the OOD objects are relatively bigger and the scenes are different than city road scenes in Cityscapes. We observe that supervised Maskomaly although localises the OOD object in some cases, but does not properly segment the object. In second test image it

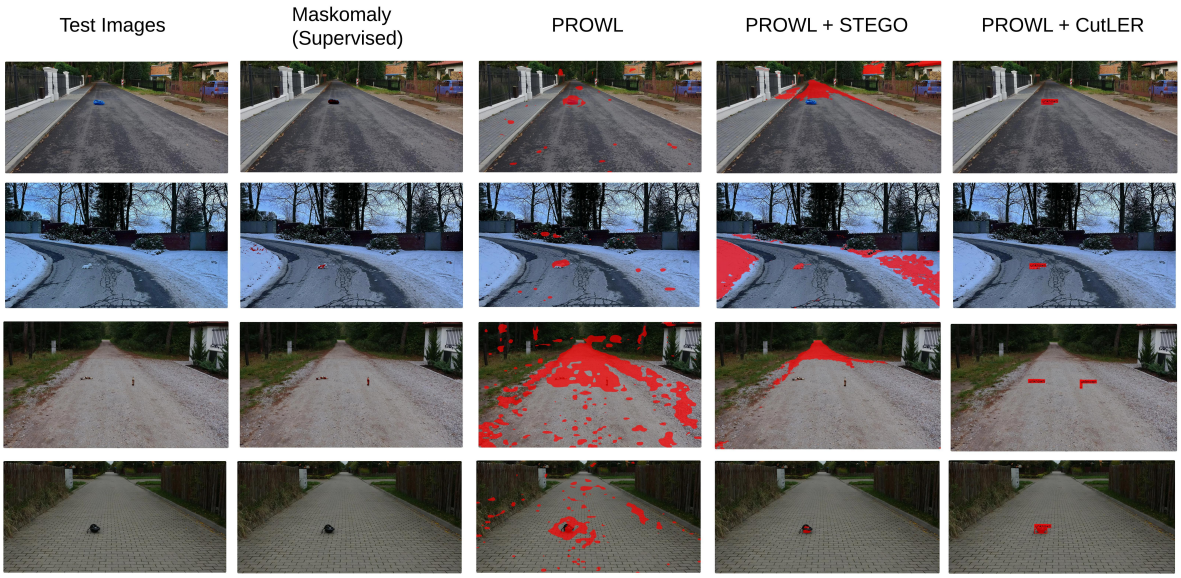


Figure 5. Performance comparison of our proposed zero-shot methods compared to supervised baseline for OOD object detection and segmentation on the test images of RoadObstacle (SMIYC- Obstacle Track) OOD dataset. Detected OOD pixels are shown in red.

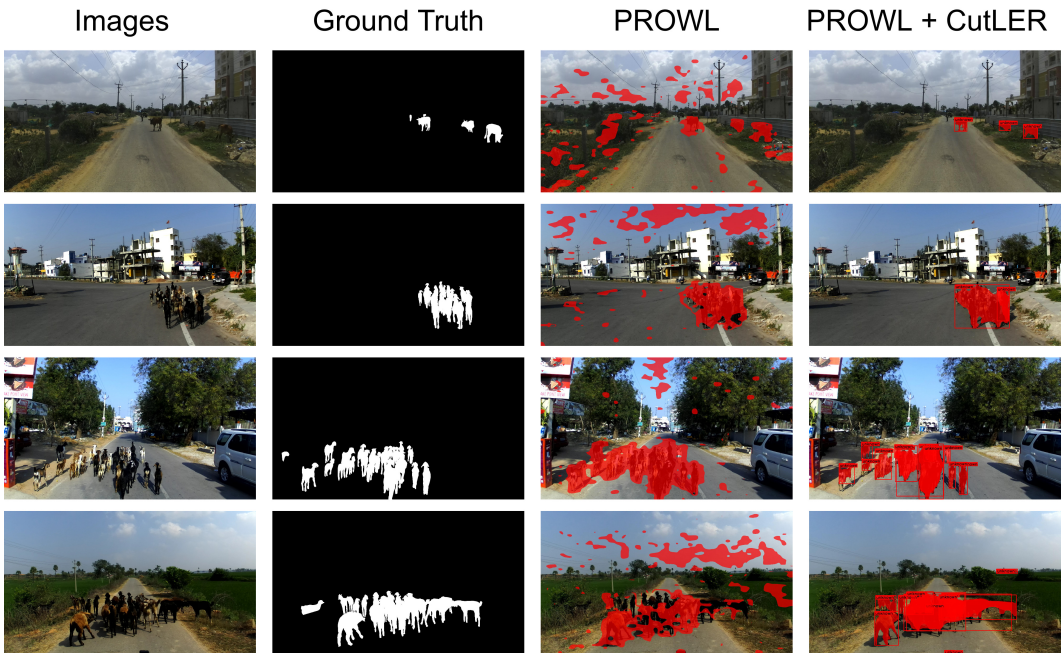


Figure 6. Qualitative performance comparison of our proposed zero-shot methods for OOD object detection and segmentation on the images from the Indian Driving Dataset [5]. Detected OOD pixels are shown in red.

falsely predicts *traffic sign* as OOD while in the third image, it misses the *dressed-up bear* as OOD object. PROWL and PROWL + STEGO localises all the OOD objects, however provides noisy segmentation including background pixels. PROWL + CutLER shows overall best performance with correctly localising and segmenting all the OOD objects.

Fig. 5 show performance comparison on the RoadObstacle dataset where the OOD objects are varying in sizes as well as the scenes in the test data show different weather conditions and different road types such as *dark asphalt, gravel, paved* and so on. This is the most challenging dataset where most methods have difficulty in spotting

small OOD objects lying very far away in diverse scenes. We observe that supervised Maskomaly localises the OOD objects in the first two test images, however fails to detect them in the last two images. PROWL and PROWL + STEGO show noisy detections whereas PROWL with CutLER localises and segments all the instances of OOD objects quite well.

Fig. 6 shows zero-shot performance of our methods on a subset of Indian Driving Dataset (IDD) [5]. IDD can easily be deemed as one of the most difficult datasets for the autonomous driving scene understanding, due to extensive traffic, crowds, and, non-regular structures on the side of the roads like different types of buildings, banners, heaps etc. Also, the presence of uncommon obstacles such as animals coming into sudden proximity of the vehicles on the road are expected to be quite a domain shift as compared European urban driving dataset such as Cityscapes. Thus, this dataset is one of the most challenging datasets for evaluating the performance of a model for OOD detection and segmentation. Since OOD objects are not explicitly specified in this dataset, we create a small OOD test subset of 20 samples containing object classes, such as *animals* which do not overlap with Cityscapes domain classes. Thus, using the prototype feature bank based on Cityscapes, we evaluate the zero-shot performance of our methods using the generic threshold of $0.55$ for INCS and $0.2$ for CutLER without requiring to fine-tune any threshold on the datasets. PROWL shows its efficacy in determining the pixel regions where the OOD objects might be present. Moreover, when we incorporate the CutLER together with, we get a more accurate OOD localization which helps in robust OOD detection and segmentation. In all sample images showing multiple instances of *animals on the road* are accurately segmented. The quantitative performance of PROWL shows an average IOU value of $26.46$, and F1 value of $39.84$ over the dataset, and PROWL+CutLER has an average IOU value of $55.99$, and F1 value of $67.47$ respectively. We note there are other fine-grained objects appearing in the scene which often get detected as OOD, although they are not deemed so nor they are present in Cityscapes ODD list.

We note that possible cases of failure often appear when the images are too dark and foreground objects in the images are not sufficiently visible.

Overall, we show that PROWL with CutLER can be readily used for plug-and-play zero-shot inference without further training or fine-tuning on the domain data, which works well for both instance segmentation on ID datasets as well as OOD detection on OOD datasets as an zero-shot method which performs comparably and also outperforms SOTA supervised methods for some OOD datasets.

## References


[1] Jan Ackermann, Christos Sakaridis, and Fisher Yu. Maskomaly:Zero-Shot Mask Anomaly Segmentation, Aug. 2023. 2

[2] 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. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, Las Vegas, NV, USA, June 2016. IEEE. 1, 2

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, Paris, France, Oct. 2023. IEEE. 1

[5] Girish Varma, Anbumani Subramanian, Anoop Namboodiri, Manmohan Chandraker, and CV Jawahar. Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1743–1751. IEEE, 2019. 4, 5

[6] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, and Zhibo Chen. Inpaint Anything: Segment Anything Meets Image Inpainting, Apr. 2023. 1

[7] Oliver Zendel, Markus Murschitz, Marcel Zeilinger, Daniel Steininger, Sara Abbasi, and Csaba Beleznai. RailSem19: A Dataset for Semantic Rail Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 3