IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Panoptic Out-of-Distribution Segmentation

Rohit Mohan, Kiran Kumaraswamy, Juana Valeria Hurtado, Kürsat Petek, and Abhinav Valada

***Abstract*—Deep learning has led to remarkable strides in scene understanding with panoptic segmentation emerging as a key holistic scene interpretation task. However, the performance of panoptic segmentation is severely impacted in the presence of out-of-distribution (OOD) objects i.e. categories of objects that deviate from the training distribution. To overcome this limitation, we propose panoptic out-of-distribution segmentation for joint pixel-level semantic in-distribution and out-of-distribution classification with instance prediction. We extend two established panoptic segmentation benchmarks, Cityscapes and BDD100K, with out-of-distribution instance segmentation annotations, propose suitable evaluation metrics, and present multiple strong baselines. Importantly, we propose the novel PoDS architecture with a shared backbone, an OOD contextual module for learning global and local OOD object cues, and dual symmetrical decoders with task-specific heads that employ our alignment-mismatch strategy for better OOD generalization. Combined with our data augmentation strategy, this approach facilitates progressive learning of out-of-distribution objects while maintaining in-distribution performance. We perform extensive evaluations that demonstrate that our proposed PoDS network effectively addresses the main challenges and substantially outperforms the baselines. We make the dataset, code, and trained models publicly available at http://pods.cs.uni-freiburg.de.**

***Index Terms*—Deep Learning for Visual Perception; Computer Vision for Transportation; Data Sets for Robotic Vision**

I. Introduction

RECENT advances in deep learning have substantially improved the capabilities of autonomous systems to interpret their surroundings. Central to these advancements is panoptic segmentation [1], which integrates semantic segmentation with instance segmentation, providing a holistic understanding of the environment. Given the potential consequences of autonomous systems malfunctioning due to unexpected inputs, it is crucial to ensure safe and robust deployment. However, existing panoptic segmentation models yield overconfident predictions of object categories out of the distribution they were trained on, known as out-of-distribution (OOD) objects. Segmenting these OOD objects poses a major challenge due to the absence of explicit knowledge about diverse OOD object characteristics, e.g., they can vary significantly in appearance and semantics, include fine-grained details, and share visual characteristics with in-distribution objects, leading to ambiguity. When learning with supervised OOD data, where training data is limited and does not encompass all OOD object variations, models can overfit to specific OOD objects encountered during training [2]. This becomes more challenging with the simultaneous identification and segmentation of OOD objects and in-distribution classes. The increased complexity makes naive adaption of methods from the less complex tasks such as semantic out-of-distribution segmentation vulnerable to trade-offs prioritizing one aspect over the other.

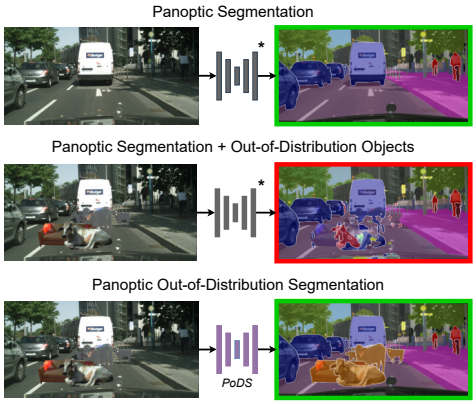


Fig. 1: The *panoptic segmentation* network (*) presents erroneous predictions when the input contains objects that are not representative of the distribution it was trained on. *Panoptic out-of-distribution segmentation* aims to address this by predicting both semantic and instance segmentation of *stuff* and *thing* classes, while also predicting instances of unseen out-of-distribution classes.

To directly address these challenges at the task level, we introduce panoptic out-of-distribution segmentation that focuses on holistic scene understanding while effectively segmenting OOD objects. Fig. 1 illustrates our proposed task that aims to predict both the semantic segmentation of *stuff* classes and instance segmentation of *thing* classes as well as an OOD class. An object is considered OOD if it is not present in the training distribution but appears in the testing/deployment stages. This distinguishes panoptic OOD segmentation from the closely related open-set panoptic segmentation [3]. Further, panoptic OOD segmentation does not reason about the semantic differences between OOD objects since in most robotics settings, especially navigation, it is sufficient to identify OOD objects and further semantically categorizing them does not provide significant utility.

In this work, we establish two challenging benchmarks, Cityscapes-OOD and BDD100K-OOD, by extending the standard autonomous driving datasets with OOD instance segmentation annotations. We present several strong baselines by combining semantic out-of-distribution segmentation methods with a class-agnostic instance segmentation decoder or adapting open-set segmentation approaches. We also introduce a tailored Panoptic Out-of-Distribution Quality (POD-Q) metric to quantify the performance. More importantly, as a first novel approach, we propose the PoDS architecture that incorporates out-of-distribution perception ability into a panoptic segmentation network conditioned on prior knowledge of in-distribution classes. By doing so, the network avoids the

Manuscript received: October, 25, 2023; Revised January, 13, 2024; Accepted March, 1, 2024.
This paper was recommended for publication by Editor Markus Vincze upon evaluation of the Associate Editor and Reviewers' comments.
Department of Computer Science, University of Freiburg, Germany.
The Supplementary material is available at https://arxiv.org/abs/2310.11797.
This work was funded by the German Research Foundation (DFG) Emmy Noether Program grant No 468878300.
Digital Object Identifier (DOI): see top of this page.

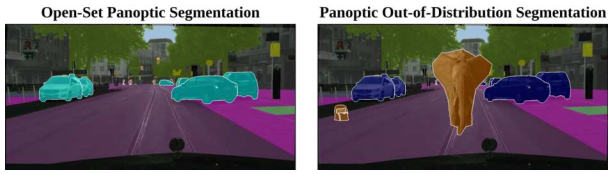


Fig. 2: Illustration of annotated samples for open-set panoptic segmentation and panoptic out-of-distribution segmentation, respectively. Cyan represents open-set classes, while orange represents the OOD class.

TABLE I: Differences between the open-set panoptic segmentation and panoptic out-of-distribution segmentation tasks, respectively.

Open-Set Panoptic Seg.	**Panoptic Out-of-Distribution Seg.**
• Categorizes classes from the same distribution within the dataset into known and open-set classes	• OoD objects are sampled from a different distribution than the underlying dataset
• Evaluates on seen objects	• Evaluates on unseen objects
• Allows overlapping categories between known and open-set classes	• Does not allow overlapping categories between known classes and OoD class
• Does not allow additional training data [4]	• Allows additional training data. Thus, accepts outlier exposure based methods.

pitfalls of ambiguously modeling both OOD and in-distribution classes, thereby improving generalization and adept handling of unseen OOD objects. We perform extensive experimental evaluations that first demonstrate the feasibility of the task and further that our proposed PoDS architecture significantly outperforms the baselines, ensuring a balanced performance on both in-distribution and out-of-distribution classes.

In summary, the contributions of this work are as follows:

1) We introduce the novel panoptic OOD segmentation task, identifying its main challenges, along with multiple baselines, and a suitable POD-Q metric.
2) We present the Cityscapes-OOD and BDD100K-OOD benchmarks, which extend the established datasets with OOD instance segmentation annotations.
3) We propose the novel PoDS architecture that incorporates the proposed modules to embed OOD segmentation capabilities into a panoptic segmentation network leveraging conditional in-distribution priors.
4) We present comprehensive quantitative and qualitative evaluations to demonstrate the feasibility of the task and the efficacy of our proposed PoDS architecture.
5) We make the code, datasets, and models publicly available at http://pods.cs.uni-freiburg.de.

II. Related Work

In this section, we present an overview of panoptic segmentation methods, followed by out-of-distribution segmentation approaches and open-set panoptic segmentation methods.

Panoptic Segmentation methods can be categorized as top-down and bottom-up approaches. Top-down methods [5] employ task-specific heads, where the instance segmentation head predicts bounding boxes and corresponding masks for objects, while the semantic segmentation head generates dense semantic predictions for each class. The outputs from these heads are then combined using heuristic-based fusion modules [6], [5]. Conversely, bottom-up methods [7] begin with semantic segmentation and then employ various techniques [8] to group *thing* pixels together to obtain instance segmentation. Recently, Mohan *et al.* [9] introduced the PAPS architecture with a shared backbone, an asymmetrical dual-decoder, and several modules for amodal panoptic segmentation [10], which predicts both visible and occluded object segments. We base our approach on PAPS's modal variant, which perceives only visible segments as it outperforms other bottom-up methods.

Semantic Out-of-Distribution Segmentation is often addressed through the use of uncertainty estimation techniques. These methods condition segmentation outputs based on threshold scores to predict OOD objects, which becomes sensitive, as any fragmentation in predictions can result in false instance predictions. A popular method is the maximum softmax probability (MSP) [11] that uses probabilities from the softmax distribution. Following, maximum logit (MaxLogit) [12] uses the negative of the maximum unnormalized logit to deliver improved performance in semantic out-of-distribution segmentation over MSP. On the other hand, Bayesian networks generate uncertainty estimates by modeling their weights and outputs as probability distributions rather than fixed values. Various frameworks also use density estimation [13] via estimating the likelihood of samples with respect to the training distribution for addressing semantic out-of-distribution segmentation. Furthermore, [2] proposes a loss function to yield high entropy for out-of-distribution sample predictions, while [14] addresses the class imbalance in OOD data through a balanced regularization loss. [15] employs residual module to augment contextual semantic features with outlier information. The use of autoencoders on in-distribution data has also been explored to identify erroneous and less reliable reconstructions of out-of-distribution samples due to unseen patterns during training. Generative models [16] create OOD boundary samples. However, scaling this to complex, high-dimensional data like high-resolution urban images is challenging. Addressing different aspects, [17] leverages the style differences between in-distribution and out-of-distribution data whereas [18] capitalizes on the behavior of object queries that function approximately as one vs. all classifiers to propose a novel outlier scoring function. On the other hand, ODIN [19] uses temperature scaling with small adversarial perturbations on the input at test time. Unlike existing methods, PoDS employs a dual predictive heads setting to address panoptic out-of-distribution segmentation, with one head specializing in in-distribution semantic categories and the other focusing on both in-distribution and out-of-distribution objects. We propose an alignment mismatch loss to encourage learning in these heads through consensus and discord between them. Furthermore, we introduce a novel OOD contextual module specifically designed to learn discriminating features of out-of-distribution objects at both global and local levels. Additionally, we integrate a dynamic module to seamlessly incorporate these features.

Open-Set Panoptic Segmentation

There are only two approaches that have been proposed for open-set panoptic segmentation thus far. EOPSN [3] groups similar unlabeled objects across multiple inputs during training and assigns labels to unlabeled objects that are surrounded by known segments. Following, Xu *et al.* [20] use a known classification head to reject segments while employing a class-agnostic classifier to identify the segments as unknown objects or open-set objects. Fig. 2 and Tab. I highlights the key

differences between our proposed panoptic out-of-distribution segmentation task and the closely related open-set panoptic segmentation task.

III. Panoptic Out-of-Distribution Segmentation

1) Task Definition: Panoptic out-of-distribution segmentation aims to assign each pixel i of an input image to an output pair $(c_i, \kappa_i) \in (C \cup O) \times N$. Here, C denotes known semantic classes, while O represents the out-of-distribution class, such that $C \cap O = \emptyset$, and N is the total number of instances. C is further divided into *stuff* labels C^S (e.g., sidewalks) and *thing* labels C^T (e.g., pedestrians). In this task, the variable c_i can be a semantic or OOD class, and κ_i indicates the corresponding instance ID. For *stuff* classes, κ_i is not applicable.

2) Evaluation Metric: To quantify the performance, it is essential to evaluate both in-distribution and out-of-distribution performance equally. However, existing metrics such as AuROC and FPR95 solely assess out-of-distribution objects and fail to account for OOD instances and in-distribution classes. Therefore, they are inadequate for evaluating panoptic out-of-distribution segmentation comprehensively. To this end, we introduce the Panoptic Out-of-Distribution Quality (POD-Q), which builds upon the panoptic quality (PQ) metric [1]. Given the predicted object segments P and their ground truth object segments G, to compute the POD-Q metric, we first compute PQ_{out} as the PQ for the OOD class o as follows:

$$PQ_{out} = \frac{\sum_{(p,g)\in TP_o,} IoU(p,g)}{|TP_o| + \frac{1}{2}|FP_o| + \frac{1}{2}|FN_o|}, \tag{1}$$

Subsequently, we compute the PQ score for all the in-distribution semantic classes PQ_{in} as

$$PQ_{in} = \frac{1}{|C|} \sum_{c\in C} \frac{\sum_{(p,g)\in TP_c,} IoU(p,g)}{|TP_c| + \frac{1}{2}|FP_c| + \frac{1}{2}|FN_c|}, \tag{2}$$

where C is the set of in-distribution semantic classes.

For both PQ_{out} and PQ_{in}, the true positives (TP_i), false positives (FP_i), and false negatives (FN_i) are defined as

$$TP_i = \{p_i \in \{P\} \mid IoU(p_i, g_i) > 0.5,\ \forall\ g_i \in \{G\}\}, \tag{3}$$
$$FP_i = \{p_i \in \{P\} \mid IoU(p_i, g) <= 0.5,\ \forall\ g \in \{G\}\}, \tag{4}$$
$$FN_i = \{g_i \in \{G\} \mid IoU(g_i, p) <= 0.5,\ \forall\ p \in \{P\}\}. \tag{5}$$

where i takes the value of o for OOD class and c for in-distribution semantic classes with $c \in C$. Finally, we compute POD-Q as the geometric mean between PQ_{out} and PQ_{in}, as

$$POD\text{-}Q = (PQ_{out} \times PQ_{in})^{\frac{1}{2}}. \tag{6}$$

We use the geometric mean to incentivize balanced performance in both out-of-distribution and in-distribution segmentation while strictly penalizing methods that only excel in one aspect of the task.

3) Baselines: We present nine baselines for the panoptic out-of-distribution segmentation task. We adapt four effective semantic out-of-distribution segmentation methods (MSP [11], MaxLogit [12], ODIN [21], Meta-OOD [2]), BE [14] with the PAPS [9] modal panoptic segmentation architecture (PAPS* as described in Sec. V-1). We compute the OOD semantic class from the semantic segmentation output from PAPS* and then use the post-processing approach from [7] for *thing+OOD* foreground segmentation to obtain the final panoptic out-of-distribution segmentation prediction. For the two remaining baselines, EPSON [3] and DD-OPS [20], we restrict the segmentation of unknown classes to a single *OOD* class and enhance their base network with EfficientPS [5], a state-of-the-art top-down panoptic segmentation network.

(a) Random scaling (b) Depth-based scaling

Fig. 3: Incorporation of OOD objects into a scene from the Cityscapes dataset is shown, with (c) random scaling and (d) depth-based OOD object scaling.

IV. Dataset Generation

Given the high cost and complexity of annotating panoptic segmentation data, it is impractical to manually label a new dataset that encompasses a diverse set of real-world out-of-distribution instances for panoptic out-of-distribution segmentation. As an alternative, we extend established urban scene understanding datasets for panoptic segmentation by incorporating real-world OOD object instances, creating two new datasets: Cityscapes-OOD and BDD100K-OOD.

We extract atypical objects from the LVIS [22] instance segmentation dataset using their segmentation masks. Objects such as cats and desks that are not present in the original panoptic segmentation dataset are added to the images. Their positions and the number of instances are randomized, with the object likelihood based on their typical locations (e.g., couches at the bottom, airplanes at the top). We further employ depth-dependent scaling to resize the OOD objects, ensuring that objects near the ego-car are relatively larger than the ones positioned far away. To do so, we begin by determining the sizes of objects in the original panoptic segmentation dataset and then match these sizes to bins established based on depth. Based on their size, the extracted OOD objects are paired with known semantic classes (e.g., surfboard with person, couch with car). We then overlay these objects, selecting a size from the depth bin randomly based on their positioning (Fig. 3 (a) and (b)). We use blending techniques [13] such as color shifts, depth blur, color curve, and gamma transformations, and we remove low-quality samples to enhance the dataset's quality and remove low-quality samples to improve the dataset's quality. Lastly, we ensure OOD objects in the training set (Fig. 4) are distinct from those in the test set (Fig. 5), guaranteeing novelty during testing, and consistent with the requirements of the panoptic out-of-distribution segmentation task.

1) Cityscapes-OOD: We create the Cityscapes-OOD dataset for panoptic out-of-distribution segmentation, with 11 *stuff* classes, 8 *thing* classes, and an *OOD* class by extending Cityscapes [23]. It consists of 2975 training and 500 test images at a resolution of 2048×1024 pixels. Test set annotations which

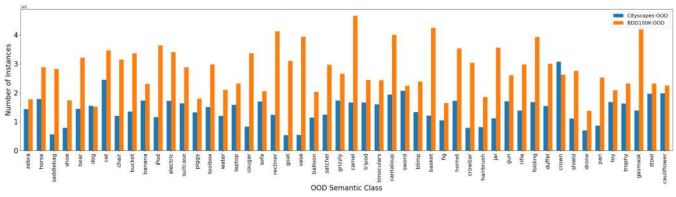


Fig. 4: OOD semantic class distribution from LVIS for the train set of Cityscapes-OOD and BDD100K-OOD datasets. Best viewed at $\times 8$ zoom.

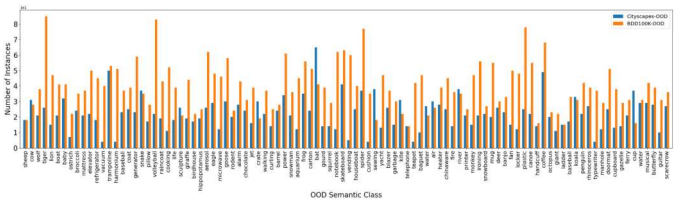
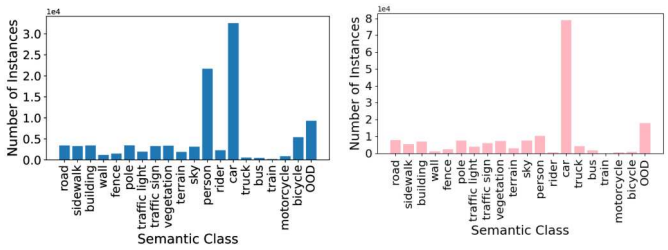


Fig. 5: OOD semantic class distribution from LVIS for the test set of Cityscapes-OOD and BDD100K-OOD datasets. Best viewed at $\times 8$ zoom.

(a) Cityscapes-OOD (b) BDD100K-OOD

Fig. 6: Sample images from the Cityscapes-OOD and BDD100K datasets.



(a) Cityscapes-OOD dataset statistics. (b) BDD100K-OOD dataset statstics.

Fig. 7: Dataset statistics for (a) Cityscapes-OOD and (b) BDD100K-OOD. Note that each *stuff* class has a single occurrence per image.

are generated from the validation set of Cityscapes are not publicly released, and evaluation is only possible through an online server. Fig. 6 (a) and Fig. 7 (a) shows an example and dataset statistics.

2) BDD100K-OOD: The *BDD100K-OOD* dataset consists of 7000 training and 1000 validation images with a resolution of 1280×720 pixels and is an extension of BDD100K [24], augmented with out-of-distribution objects. It features one *OOD* class, 11 *stuff* classes including roads and buildings, and eight *thing* classes such as cars and bicycles. Fig. 6 (b) and Fig. 7 (b) present an example and dataset statistics, respectively.

V. PoDS Network Architecture

In this section, we detail our proposed PoDS architecture depicted in Fig. 8. We first present an overview of the PoDS network, followed by a detailed description of each constituting component. PoDS builds on top of a base panoptic segmentation network that has a shared backbone and task-specific decoders (purple) by incorporating modules specially designed to embed out-of-distribution capabilities based on prior knowledge of in-distribution classes. We incorporate an OOD contextual module (blue) that complements the robust in-distribution semantic features of the shared backbone with both global discriminatory and fine local OOD object representations. Subsequently, we introduce an additional task-specific decoder (green), equipped with dynamic modules, alongside the existing ones. This design allows for adaptive integration of OOD features while preserving the in-distribution features of the high-performing base panoptic network. The unique dual task-specific decoder configuration benefits further from our novel alignment-mismatch loss. This loss encourages learning finer details within in-distribution semantic classes and what lies outside by balancing consensus and divergence between the two heads. Furthermore, we incorporate a data augmentation strategy to facilitate the training of our novel modules. Please refer to Sec. S.1 of the supplementary material for further implementation details.

1) Base Network: Building on the modal variant of PAPS [9], which excels in panoptic segmentation, we develop an architecture for panoptic out-of-distribution segmentation. The modal PAPS architecture has a shared backbone, decoders, prediction heads, a context extractor, and a cross-task module. The shared backbone generates four parallel feature map outputs at scales $\times 4$, $\times 8$, $\times 16$, and $\times 32$ with respect to the input, named B_4, B_8, B_{16}, and B_{32}. Following, the outputs from the backbone are fed to both the semantic and instance decoder. For our PoDS network, we streamline PAPS architecture by excluding the instance segmentation decoder and cross-task module. Instead, we adopt the semantic segmentation decoder with the dense prediction cell (DPC) module, along with two upsampling stages ($\times 8$ and $\times 4$) and skip connections for the instance segmentation decoder. The instance segmentation head remains intact, handling instance center prediction and regression. This streamlined PAPS architecture, termed PAPS*, achieves a PQ score of 63.7 on the Cityscapes validation set, close to PAPS's 64.3. As shown in Fig. 8 purple boxes with red locks, we pretrain PAPS* on in-distribution panoptic segmentation datasets, and keep its weights fixed throughout the out-of-distribution segmentation training.

2) OOD Contextual Module: We introduce the OOD Contextual Module for the PoDS architecture, designed to capture both global and local features of out-of-distribution (OOD) objects in images. As depicted in Fig. 8 (blue box), this module incorporates two residual bottleneck blocks, similar to the fourth (scale 16) and fifth stages (scale 32) of Regnet [25]. The module takes the output from the last layer of the backbone's stage 2 (scale 8), processes it through the first block, combines it with the output from the last layer of stage 3 (scale 16), and routes it through the second block. Subsequently, the output from the second block, named $\mathrm{O}_{\mathrm{ocm}}$, proceeds to a global average pooling layer and then to a classification head. In parallel, $\mathrm{O}_{\mathrm{ocm}}$ undergoes upsampling by a factor of 4 followed by two convolution layers. This processed output splits into two branches: one path goes to an in/out-distribution segmentation head, while the other undergoes further upsampling by a factor of 2 and convolutions before reaching a second segmentation head. Both heads distinguish between pixel-wise in-distribution

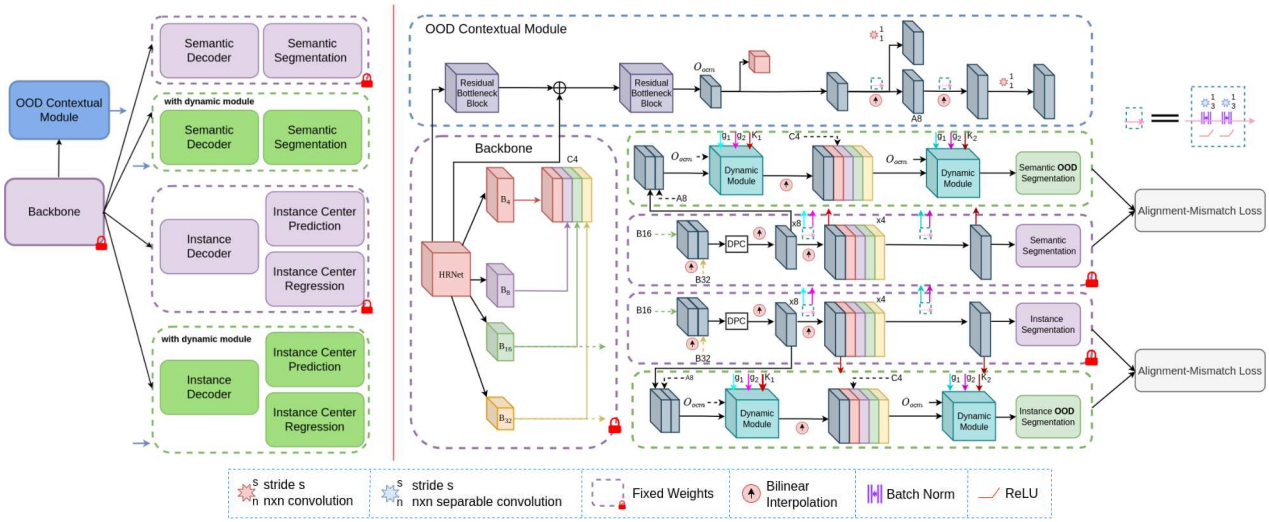


Fig. 8: Illustration of our proposed PoDS architecture that consists of a shared backbone with an OOD contextual module and symmetrical task-specific decoder arranged in a dual configuration setup to facilitate an alignment-mismatch learning strategy. The shared backbone learns robust feature representations for in-distribution semantic categories while the OOD contextual module supports both global and local features for OOD objects. The network comprises symmetrical semantic and instance decoders that include dynamic modules to adaptively balance the features between in- and out-distribution representations.

and out-of-distribution regions. During training, we apply random data augmentation, as detailed in Sec. VI-A, generating samples with OOD objects from the panoptic segmentation dataset to train both the classification (targeting OOD global features) and pixel-wise segmentation heads (focusing on OOD local features). We employ binary cross-entropy loss for the classification head and weighted pixel-wise cross-entropy loss for the in/out-distribution segmentation heads, with weights of 0.7 and 0.8 respectively. Thus, L_{ocm} is the sum of the aforementioned losses. Notably, backpropagation for the pixel-wise loss is only triggered when an OOD object is present in the input.

3) Dynamic Module: The dynamic module is defined by the following inputs: an input feature map F, O_{ocm} from the OOD contextual module, and feature map and convolution functions (K and $g_1(\cdot, w_1)$ and $g_2(\cdot, w_2)$, respectively) from the base network. The inputs from the base network are represented by the red, yellow, and pink arrows in Fig. 8, respectively.

$$F_{I_1} = g_1^*(F, w_1 + \Delta w_1^*), \tag{7}$$
$$F_{I_2} = g_2^*(F_{I_1}, w_2 + \Delta w_2^*), \tag{8}$$
$$F_O = h_1(O_{ocm}) \cdot F_{I_2} + (1 - h_1(O_{ocm})) \cdot K, \tag{9}$$

where $g_1^*(\cdot, w_1 + \Delta w_1^*)$ and $g_2^*(\cdot, w_2 + \Delta w_2^*)$ denote convolution functions with learned weights w_1^* and w_2^* as offsets from the weights w_1 and w_2 of $g_1(\cdot, w_1)$ and $g_2(\cdot, w_2)$, respectively. The computation of F_{I_2},representing the intermediate features, is sequentially performed. It starts with an initial input F, which is first processed by the function $g_1^*(\cdot)$, yielding F_{I_1}. This output F_{I_1}, is then further processed by the function $g_2^*(\cdot)$ to obtain F_{I_2}. h_1 is the weighted gating function, composed of a consecutive global pooling followed by a 1×1 convolution layer. It takes the discriminative out-of-distribution features from OOD contextual module O_{ocm} as input. This gating function enables adaptive fusion of the base network's features K with the intermediate feature F_{I_2}, yielding the final output F_O of the module.

4) PoDS Decoders and Heads: In the PoDS framework, we utilize additional decoders akin to the base network described in Sec. V-1, as visualized with green boxes in Fig. 8. Each decoder starts by merging the upsampled DPC features from their base network's task-specific decoders (purple boxes) with the A_8 features from the OOD contextual module (blue boxes). The resulting features (F) are then processed by a dynamic module, which also takes in the output of the existing convolution layers (K_1) of the $\times 8$ stage in the base network. The output of the module (F_O) is then upsampled and concatenated with C_4 (Sec. V-1) and both are fed to another dynamic module along with the output of the existing convolution layers (K_2) of $\times 4$ stage in the base network. The final output of this module is then fed to the corresponding task-specific heads.

The PoDS base network targets only in-distribution classes. To expand the network's capabilities, we incorporate additional task-specific heads that can learn about both in-distribution and out-of-distribution classes. These heads consist of two sequential layers of 3×3 depthwise-separable convolutions, followed by a task-specific 1×1 predictor. The *OOD* semantic segmentation head uses a predictor with $N_{stuff} + N_{thing} + 1$ for segmentation labels. The *OOD* instance segmentation heads have two predictors: instance center prediction and instance center regression, which learn on *thing* + *void* regions. To train the semantic head, we use the weighted bootstrapped cross-entropy loss (L_{sem}). For the instance center prediction, we use the Mean Squared Error (MSE) loss (L_{center}) to minimize the distance between the predicted heatmaps and the 2D Gaussian encoded groundtruth heatmaps. For instance center regression, we use the $L1$ loss (L_{reg}).

5) Learning from alignment-mismatch: We train the semantic segmentation head SH_{in} from the PoDS base network only for known in-distribution classes and we train the PoDS head SH_{out} for an added *OOD* class. The SH_{in} consistently labels pixels with known semantic classes, irrespective of in- or out-of-distribution object. During the training of SH_{out}, we aim to amplify the prediction discrepancies between

(SH_{in} and SH_{out}) for out-of-distribution class pixels while promoting consensus for in-distribution object predictions. To implement the alignment-mismatch strategy, we ensure the output dimensions of SH_{in} and SH_{out} match. Given SH_{in} has $(N_{stuff} + N_{thing}) \times H \times W$ channels and SH_{out} has $(N_{stuff} + N_{thing} + 1) \times H \times W$, we derive an extra channel for SH_{in} by taking the maximum across the semantic class dimension. We employ the following loss to foster alignment-mismatch between the two heads, depending on whether the pixel belongs to an in-distribution or out-of-distribution object:

$$e_i = |s(SH_{in}(x_i)) - s(SH_{out}(x_i))|^2, \quad (10)$$

$$k_i = (1 - y_i)e_i, \quad (11)$$

$$d_i = y_i \max(0, m - e_i), \quad (12)$$

$$L_{sem-am} = \frac{1}{2N} \sum_{i=1}^{N} k_i + d_i, \quad (13)$$

where N is the number of pixels, x_i is the input image pixel, y_i is the label that indicates out- or in-distribution class, m is the hyperameter and $s = \ln(1 + e^x)$ is the softplus acitvation function. We use the softplus to encourage that SH_{out} predicts void class for out-of-distribution pixels with higher logits compared to SH_{in}'s maximum logit. Since softplus always yields positive output and weights of SH_{in} are frozen, the only way for SH_{out} to reduce the loss is by predicting out-of-distribution classes with high logits, especially when the margin hyperparameter m in the loss is sufficiently large.

For instance segmentation, we strive to foster alignment-mismatch between the instance center prediction and instance center regression heads. We achieve this by employing a similar loss as L_{sem-am} but applied in the feature space to the features X_{in} and X_{out} prior to the predictor for respective heads. This separates in-distribution from out-of-distribution features, making it easier to perform the center prediction and center regression. We define instance segmentation losses as

$$e_i = |X^i_{j-in} - X^i_{j-out}|, \quad (14)$$

$$L_{j-am} = \frac{1}{2N} \sum_{i=1}^{N} k_i + d_i, \quad (15)$$

where X^i_{j-in} and X^i_{j-out} are the features computed at location i for the instance segmentation heads. k_i and d_i are the same as (11) and (12), respectively. $j \in [center, reg]$ represents the instance center prediction or instance center regression heads, from which we obtain the losses $L_{center-am}$ and L_{reg-am}, respectively.

VI. Experimental Evaluation

A. Training and Inference Protocol

We adopt a two-stage training approach for our network. Initially, we train the base layers of the PoDS network for 160,000 iterations on Cityscapes and 240,000 iterations on BDD100K to instill strong in-distribution priors. In the second stage, these base layers are frozen and the remaining components of the PoDS network are trained for 90K and 150K iterations on Cityscapes and BDD100K, respectively. This step incorporates data augmentation techniques, where we generate training samples by combining in-distribution class samples with those that include both in- and out-of-distribution objects. To curate out-of-distribution samples, we source web images using specific keywords, ensuring they exclude known in- or out-of-distribution objects from the Cityscapes-OOD and BDD100K-OOD test set. Using an unsupervised instance segmentation network, we generate pseudo instance masks for these images, facilitating the extraction and compilation of a diverse out-of-distribution (OOD) object repository. During training, images are either augmented with randomly positioned and scaled OOD objects or left as in-distribution. For each training phase, we employ the Adam optimizer with a poly learning rate schedule, setting the initial learning rates at 0.001 for Cityscapes and 0.005 for BDD100K. We optimize the following loss functions for training the network:

$$L = L_{ocm} + L_{sem} + \alpha L_{center} + \beta_1(L_{reg} + L_{sem-am}) + \beta_2(L_{center-am} + L_{reg-am}), \quad (16)$$

where $\alpha = 200$, $\beta_1 = 0.01$ and $\beta_2 = 0.001$. All of the individual losses are defined in Sec. V. We set the margin hyperparameter m to 50. During inference, we use the same post-processing as [7] with the semantic and instance segmentation head predictors that learn with the inclusion of OOD class.

TABLE II: Panoptic out-of-distribution benchmarking results on the Cityscapes-OOD and BDD100K-OOD test set. Subscripts *out* and *in* refer to out-of-distribution class and in-distribution classes, respectively. All scores in [%].

Model	Cityscapes-OOD			BDD100K-OOD		
	POD-Q	PQ_{out}	PQ_{in}	POD-Q	PQ_{out}	PQ_{in}
MSP [11]	12.8	3.4	47.6	9.1	2.6	32.1
MaxLogit [12]	15.9	5.2	48.6	12.7	4.7	34.5
ODIN [19]	20.8	8.7	49.8	16.9	7.9	36.1
EPSON [3]	29.4	15.9	54.4	23.7	14.3	39.4
Meta-OOD [2]	41.7	31.3	55.6	34.5	28.6	41.6
BE [14]	44.9	38.6	52.3	37.2	36.1	38.5
RbA [18]	45.4	34.1	60.5	37.8	31.6	45.2
DD-OPS [20]	46.1	36.1	58.7	38.0	33.2	43.5
PoDS(Ours)	**53.4**	**45.9**	**62.2**	**42.3**	**38.7**	**46.3**

B. Benchmarking Results

In Tab. II, we compare the performance of our PoDS architecture with the baselines on the Cityscapes-OOD and BDD100K-OOD test sets. The first three baselines, MSP [11], MaxLogit [12], and ODIN [19], adapt any panoptic segmentation network for out-of-distribution segmentation without modifications. We observe that they perform poorly compared to other reported methods. While thresholding confidence scores from these baselines enhances OOD object sensitivity, it often results in fragmented detections and misclassifications of in-distribution objects as OOD. Consequently, these baselines are not ideal for directly employing them for panoptic out-of-distribution segmentation. EPSON [3] mines labels from void regions to learn clusters for OOD objects. While it improves OOD detection and reduces in-distribution misclassification, its low POD-Q scores indicate limited generalization to unseen *OOD* objects during testing. Meta-OOD [2] prioritizes higher entropy for OOD

TABLE III: Evaluation of various architectural components in PoDS. Results are presented on the Cityscapes-OOD test set. Subscripts out and in refer to out-of-distribution class and in-distribution classes. All scores are in [%].

Model	POD-Q	PQ_{out}	PQ_{in}
M1 (PAPS*)	28.2	16.2	49.3
M2 (M1 + dual predictive head)	26.9	15.3	47.5
M3 (M2 + OOD contextual module)	47.1	38.4	58.0
M4 (PoDS, M3 + dynamic module)	**53.4**	**45.9**	**62.2**

predictions. In contrast, BE [14], by focusing on addressing the imbalance in the OOD class, achieves the highest PQ_{out} among the baselines. However, it falls short in PQ_{in} compared to other baselines. DD-OPS [20] enhances the use of void regions, rejecting objects identified with known classes and utilizing a class-agnostic classifier for OOD determination. Adopting a similar school of thought, RbA [18] utilizes one-vs-all classifiers, and both models achieve comparable performance. PoDS employs a dual-head predictive setting to delineate the boundaries between in-distribution and out-of-distribution classes and to increase confidence in OOD object segmentation during training. By leveraging the alignment and mismatch between the heads, PoDS achieves the highest POD-Q scores of 53.4 on Cityscapes-OOD and 42.3 on BDD100K-OOD.

C. Ablation Study

1) Detailed Study on the PoDS Architecture: While developing the PoDS architecture, we incorporated various components to address specific challenges. Tab. III presents four model configurations, labeled M_i, to determine the impact of each component. The M1 configuration uses the base network PAPS* with OOD classes as an extra class, trained from scratch with data augmentation. We observe that M1 achieves a low POD-Q score of 28.2 as it tries to cover both in-distribution and out-of-distribution classes, leading to poor generalization on the test set for unseen *OOD* classes. In M2, we incorporate the dual predictive head architecture of PoDS into M1 and train it with our alignment-mismatch loss. However, this leads to a drop of 1.3 in the POD-Q score compared to M1, indicating that the pretrained backbone does not encode OOD objects effectively enough for the new decoders and heads to learn. As a result, the alignment-mismatch loss hinders the performance of M2. In M3, we incorporate the OOD contextual module into M2. The notable performance improvement compared to M2 indicates that by learning highly discriminative features through the simplified task of OOD classification and segmentation, the dual predictive head, combined with the alignment-mismatch loss, prioritizes understanding what lies outside the in-distribution rather than trying to model the distribution of out-of-distribution objects.

Finally, in M4, we incorporate the dynamic module into the non-pretrained decoders of M3. The results of M4, with a POD-Q score of 50.5, underscore the benefits of learning features not previously encompassed in the in-distribution feature space. As elaborated in Sec. V-3, this is achieved by leveraging pretrained weight offsets and dynamically transitioning between the robust knowledge base of the pretrained decoder for in-distribution classes and the decoder trained to recognize features in the presence of both in-distribution and out-of-distribution objects. Moreover, the improvements from M_{i-1} to M_i result not only from new additions but also from their synergy with the existing modules. M4 embodies our proposed PoDS architecture.

TABLE IV: Evaluation of best panoptic out-of-distribution segmentation methods on the Cityscapes val set. Subscript out and in refers to out-of-distribution and in-distribution classes. Subscript $base$ refers to the base panoptic segmentation network. All scores are in [%].

(a) Panoptic out-of-distribution performance using bicycle and motorcycle classes as OOD.

Model	POD-Q	PQ_{out}	PQ_{in}
Meta-OOD [2]	39.1	27.1	56.3
DD-OPS [20]	44.7	33.4	59.8
PoDS	50.3	39.8	63.6

(b) Influence of OOD seg. on in-distribution performance.

Model	PQ	PQ_{base}
Meta-OOD [2]	60.7	63.9
DD-OPS [20]	62.5	63.9
PoDS	63.1	63.7

TABLE V: Performance of PoDS models trained on Cityscapes but evaluated on BDD100K val set and BDD100K-OOD test set. All scores are in [%].

Training Dataset	Method	Evaluation Dataset			
		BDD100K	BDD100K-OOD		
		PQ	POD-Q	PQ_{in}	PQ_{out}
Cityscapes	PAPS*	39.6	–	–	–
	PoDS	38.9	28.1	38.2	20.6

2) Evaluation in Real-World OOD Scenarios: In this experiment, we evaluate the utility of our models and baselines in real-world settings using the Cityscapes dataset. We include two *thing* classes, bicycle, and motorcycle, from the eight Cityscapes *thing* classes as part of the OOD class. We exclude any image from the training set containing at least one instance of this OOD class, reducing the training set from 2975 to 2620 images. This exclusion ensures that the bicycle and motorcycle classes are treated as unseen OOD objects during evaluation. The results, presented in Tab. IVa demonstrate that PoDS consistently outperforms the top two baselines by a substantial margin, reinforcing the findings from Tab. II and underscoring its applicability to real-world OOD scenarios. In Sec. S.2, we further qualitatively demonstrate the generalization ability of PoDS in real-world driving scenarios using our in-house data collected in Freiburg.

3) Influence of OOD Segmentation on In-Distribution Performance: We first study the impact of learning panoptic out-of-distribution segmentation on network performance when only in-distribution classes are present in the input. We compare with three methods: Meta-OOD, DD-OPS, and PoDS, and also report the performance of their base panoptic segmentation networks. From the results shown in Tab. IVb, we observe that the PQ score of Meta-OOD substantially decreases, while DD-OPS and PoDS show a smaller drop in performance. However, PoDS shows the least drop of 0.6, demonstrating the ability to segment out-of-distribution objects while preserving the in-distribution class knowledge. Subsequently, we evaluate the generalization ability of PoDS by training it on the Cityscapes dataset and evaluating it on the BDD100K dataset. Results from this experiment presented in Tab. V show that PoDS performs nearly as well as its base network PAPS*, achieving a POD-Q score of 28.1 on BDD100K-OOD and a PQ of 38.9 on BDD100K. This highlights the ability of PoDS to infer known semantic class boundaries from Cityscapes, constrained only by its base network's performance. We anticipate further advancements in this field by the robotics community will surpass these limitations in the future.

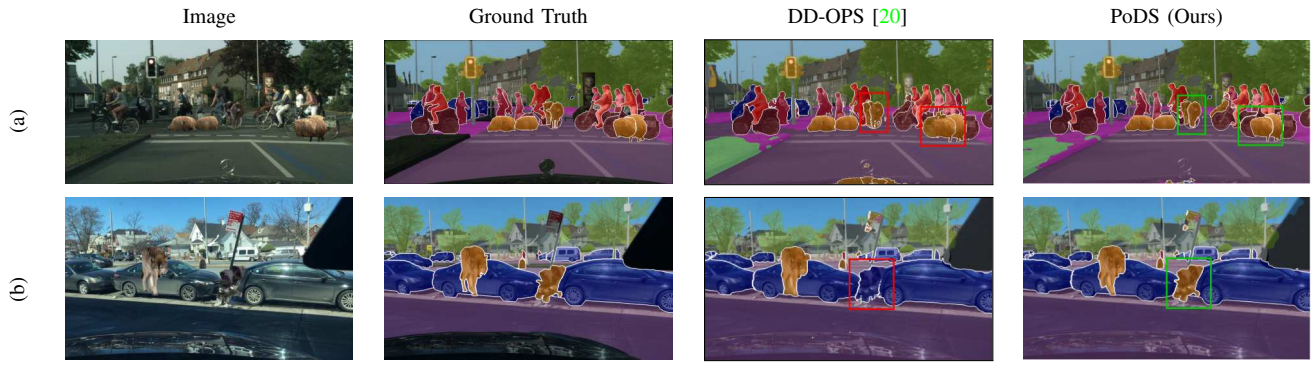


Fig. 9: Qualitative panoptic out-of-distribution segmentation results of our proposed PoDS network in comparison to the state-of-the-art baseline DD-OPS [20] on (a) Cityscapes-OOD and (b) BDD100K-OOD datasets. Red boxes highlight regions with errors in the segmentation, while the corresponding green boxes indicate regions with correct segmentation.

D. Qualitative Evaluations

We qualitatively compare the performance of PoDS with the best-performing baseline DD-OPS [20] as illustrated in Fig. 9. We observe that DD-OPS misclassifies OOD objects with known semantic classes, while PoDS excels at distinguishing them. PoDS exploits its dynamic module and the alignment-mismatch strategy to identify OOD features based on known semantic characteristics, enabling it to accurately distinguish between OOD objects, and bicycles and cars.

VII. Conclusion

In this work, we introduced the panoptic out-of-distribution segmentation task, proposed two suitable datasets, established an interpretable evaluation metric, and adapted several open-set and semantic out-of-distribution segmentation methods for baselines. We also proposed the novel PoDS architecture, which sets a new benchmark in performance. It also demonstrates the feasibility of incorporating OOD segmentation without a significant drop in in-distribution performance. We presented an extended evaluation of each module in our network with quantitative and qualitative evaluations that demonstrate their utility. Our novel framework shows the feasibility of this crucial and holistic scene parsing task and we aim that our publicly released datasets and benchmark facilitate further research.

References

[1] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.
[2] R. Chan, M. Rottmann, and H. Gottschalk, "Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 5128–5137.
[3] J. Hwang, S. W. Oh, J.-Y. Lee, and B. Han, "Exemplar-based open-set panoptic segmentation network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2021, pp. 1175–1184.
[4] T. E. Boult, S. Cruz, A. R. Dhamija, M. Gunther, J. Henrydoss, and W. J. Scheirer, "Learning and the unknown: Surveying steps toward open world recognition," in *Proc. of the AAAI conf. on artificial intelligence*, vol. 33, no. 01, 2019, pp. 9801–9807.
[5] R. Mohan and A. Valada, "Efficientps: Efficient panoptic segmentation," *Int. Journal of Computer Vision*, vol. 129, no. 5, pp. 1551–1579, 2021.
[6] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.
[7] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020.
[8] J. Uhrig, E. Rehder, B. Fröhlich, U. Franke, and T. Brox, "Box2pix: Single-shot instance segmentation by assigning pixels to object boxes," in *IEEE Intelligent Vehicles Symposium*, 2018, pp. 292–299.
[9] R. Mohan and A. Valada, "Perceiving the invisible: Proposal-free amodal panoptic segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9302–9309, 2022.
[10] ——, "Amodal panoptic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2022, pp. 21 023–21 032.
[11] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, 2017.
[12] D. Hendrycks, S. Basart, M. Mazeika, A. Zou, J. Kwon, M. Mostajabi, J. Steinhardt, and D. Song, "Scaling out-of-distribution detection for real-world settings," *ICML*, 2022.
[13] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, and C. Cadena, "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving," in *Proc. of the IEEE/CVF Int. Conf. on Computer Vision Workshops*, 2019.
[14] H. Choi, H. Jeong, and J. Y. Choi, "Balanced energy regularization loss for out-of-distribution detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2023.
[15] Y. Liu, C. Ding, Y. Tian, G. Pang, V. Belagiannis, I. Reid, and G. Carneiro, "Residual pattern learning for pixel-wise out-of-distribution detection in semantic segmentation," in *Int. Conf. on Computer Vision*, 2023.
[16] Y. Xia, Y. Zhang, F. Liu, W. Shen, and A. L. Yuille, "Synthesize then compare: Detecting failures and anomalies for semantic segmentation," in *Europ. Conf. on Computer Vision*, 2020, pp. 145–161.
[17] D. Zhang, K. Sakmann, W. Beluch, R. Hutmacher, and Y. Li, "Anomaly-aware semantic segmentation via style-aligned ood augmentation," in *Int. Conf. on Computer Vision*, 2023, pp. 4065–4073.
[18] N. Nayal, M. Yavuz, J. F. Henriques, and F. Güney, "Rba: Segmenting unknown regions rejected by all," in *Int. Conf. on Computer Vision*, 2023.
[19] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *ICLR*, 2018.
[20] H.-M. Xu, H. Chen, L. Liu, and Y. Yin, "Dual decision improves open-set panoptic segmentation," in *British Mac. Vision Conf.*, 2022.
[21] V. Besnier, A. Bursuc, D. Picard, and A. Briot, "Triggering failures: Out-of-distribution detection by learning from local adversarial attacks in semantic segmentation," in *Int. Conf. on Computer Vision*, 2021.
[22] A. Gupta, P. Dollar, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 5356–5364.
[23] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
[24] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 2636–2645.
[25] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, and Z. Xu, "Regnet: self-regulated network for image classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.