# APPENDICES

## Contents

# A. Details of the Test Sets

This section provides the details of our test sets summarized in Tab. 1. To give a general overview, while constructing our datasets, we impose restrictions for a more principled evaluation:

- We ensure that there is at least one ID object in the images of $\mathcal{D}_{\text{ID}}$ (and also in the ones in $\mathcal{T}(\mathcal{D}_{\text{ID}})$) to avoid a situation that an ID image does not include any ID object.
- $\mathcal{D}_{\text{OOD}}$ images does not include any foreground object. Besides, we use detection datasets with different ID classes (iNat, Obj365 and SVHN) than our $\mathcal{D}_{\text{Train}}$ to promote OOD objects in OOD images.

In the following, we present how we curate each of these test splits, that are (i) Obj45K and BDD45K as $\mathcal{D}_{\text{ID}}$; (ii) Obj45K-C and BDD45K-C as $\mathcal{T}(\mathcal{D}_{\text{ID}})$; and (iii) SiNObj110K-OOD as $\mathcal{D}_{\text{OOD}}$.

## A.1. Obj45K and BDD45K Splits

We construct $\mathcal{D}_{\text{ID}}$ from different but semantically similar datasets; thereby introducing domain-shift to be reflective of the challenges faced by detectors in practice such as distribution shifts over time or lack of data in a particular environment. To do so, we employ Objects365 [63] for our SAOD-Gen use-case using COCO as ID data and BDD100K for our SAOD-AV use-case with nuImages comprising the ID data. In the following, we discuss the specific details how we constructed our Obj45K and BDD45K splits from these datasets.

### A.1.1 Obj45K Split

We rely on Objects365 [63] to construct our Gen-OD ID test set. Similar to COCO [43], which we use for training and validation in our Gen-OD setting, Objects365 is a general object detection dataset. On the other hand, Object365 includes 365 different classes, which is significantly larger than the 80 different classes in COCO dataset. Therefore, using images from Objects365 to evaluate a model trained on COCO requires a proper matching between the classes of COCO with those of Objects365. Fortunately, by design, Objects365 already includes most of the classes of COCO in order to facilitate using these datasets together. However, we inspect the classes in those datasets more thoroughly to prove a more proper one-to-many matching from COCO classes to Objects365 classes. As an example, examining the objects labelled as `chair` in COCO dataset, we observe that wheelchairs also pertain to the `chair` class of COCO. However, in Objects365 dataset, `Wheelchair` and `Chair` are different classes. Therefore, in this case, we match `chair` class of COCO not only with `Chair` but also with `Wheelchair` of Objects365. Having said that, we also note that due to high numbers of images and classes in those datasets, it is not practical to have a manual inspection over all images and classes. In the following, we present our resulting matching between COCO and Objects365 classes:

```
'person':'Person',
'bicycle':'Bicycle',
'car':['Car', 'SUV','Sports␣Car',
'Formula␣1␣'],
'motorcycle':'Motorcycle',
'airplane':'Airplane',
'bus':'Bus',
'train':'Train',
'truck':['Truck','Pickup␣Truck',
'Fire␣Truck', 'Ambulance',
'Heavy␣Truck'],
'boat':['Boat', 'Sailboat', 'Ship'],
'traffic␣light':'Traffic␣Light',
'fire␣hydrant':'Fire␣Hydrant',
'stop␣sign':'Stop␣Sign',
'parking␣meter':'Parking␣meter',
'bench':'Bench',
'bird':['Wild␣Bird', 'Duck',
'Goose', 'Parrot', 'Chicken'],
'cat':'Cat',
'dog':'Dog',
'horse':'Horse',
'sheep':'Sheep',
'cow':'Cow',
'elephant':'Elephant',
'bear':'Bear',
'zebra':'Zebra',
'giraffe':'Giraffe',
'backpack':'Backpack',
'umbrella':'Umbrella',
'handbag':'Handbag/Satchel',
'tie':['Tie', 'Bow␣Tie'],
'suitcase': 'Luggage',
'frisbee':'Frisbee',
'skis':'Skiboard',
'snowboard':'Snowboard',
'sports␣ball':['Baseball', 'Soccer',
'Basketball', 'Billards',
'American␣Football', 'Volleyball',
'Golf␣Ball','Table␣Tennis␣','Tennis'],
'kite':'Kite',
'baseball␣bat':'Baseball␣Bat',
'baseball␣glove':'Baseball␣Glove',
'skateboard':'Skateboard',
'surfboard':'Surfboard',
'tennis␣racket':'Tennis␣Racket',
'bottle':'Bottle',
'wine␣glass':'Wine␣Glass',
'cup':'Cup',
'fork':'Fork',
’knife’:’Knife’,
’spoon’:’Spoon’,
’bowl’:’Bowl/Basin’,
’banana’:’Banana’,
’apple’:’Apple’,
’sandwich’:’Sandwich’,
’orange’:’Orange/Tangerine’,
’broccoli’:’Broccoli’,
’carrot’:’Carrot’,
’hot␣dog’:’Hot␣dog’,
’pizza’:’Pizza’,
’donut’:’Donut’,
’cake’:’Cake’,
’chair’:[’Chair’, ’Wheelchair’],
’couch’:’Couch’,
’potted␣plant’:’Potted␣Plant’,
’bed’:’Bed’,
’dining␣table’:’Dinning␣Table’,
’toilet’:[’Toilet’, ’Urinal’],
’tv’:’Moniter/TV’,
’laptop’:’Laptop’,
’mouse’:’Mouse’,
’remote’:’Remote’,
’keyboard’:’Keyboard’,
’cell␣phone’:’Cell␣Phone’,
’microwave’:’Microwave’,
’oven’:’Oven’,
’toaster’:’Toaster’,
’sink’:’Sink’,
’refrigerator’:’Refrigerator’,
’book’:’Book’,
’clock’:’Clock’,
’vase’:’Vase’,
’scissors’:’Scissors’,
’teddy␣bear’ : ’Stuffed␣Toy’,
’hair␣drier’:’Hair␣Dryer’,
’toothbrush’:’Toothbrush’
```

Having matched the ID classes, we label the remaining classes of Objects365 either as “OOD” or “ambiguous”. Specifically, a class is labelled as OOD if COCO classes (or nuImages classes that we are interested in) do not contain that class and they will be discussed in Section A.3. Subsequently, we label a class as an ambiguous class in the cases that we cannot confidently categorize the class neither as ID nor as OOD. As an example, having examined quite a few COCO images with `bottle` class, we haven’t observed a flask, which is an individual class of Objects365 (`Flask`). Still, as there might be instances of flask labelled as `bottle` class in COCO, we categorize `Flask` class of Objects365 as ambiguous and do not use any of the images in Objects that has a `Flask` object in it. Following this, we identify the following 25 out of 365 classes in Objects365 as ambiguous:

```
’Nightstand’, ’Desk’, ’Coffee␣Table’,
’Side␣Table’, ’Watch’, ’Stool’,
’Machinery␣Vehicle’, ’Tricycle’,
’Carriage’, ’Rickshaw’, ’Van’,
’Traffic␣Sign’,’Speed␣Limit␣Sign’,
’Crosswalk␣Sign’, ’Flower’, ’Telephone’,
’Tablet’, ’Flask’, ’Briefcase’,
’Egg␣tart’, ’Pie’, ’Dessert’, ’Cookies’,
’Wallet/Purse’
```

Finally, we collect 45K images for Obj45K split from validation set of Objects365 that contains (i) at least one ID object based on the one-to-many matching between classes of COCO and Objects365; and (ii) no object from an ambiguous class. Compared to COCO val set with 5K images with 36K annotated objects, our Gen-OD ID test has 45K images with 237K objects, significantly outnumbering the val set which is commonly used to analyse and test the models mainly in terms of robustness aspects. Fig. A.6(a) compares the number of objects for Obj45K split and COCO val set, showing that the number of objects for each class of our Obj45K split is for almost classes (except 2 of 80 classes) larger than the COCO val set. This large number of objects enables us to evaluate the models thoroughly.

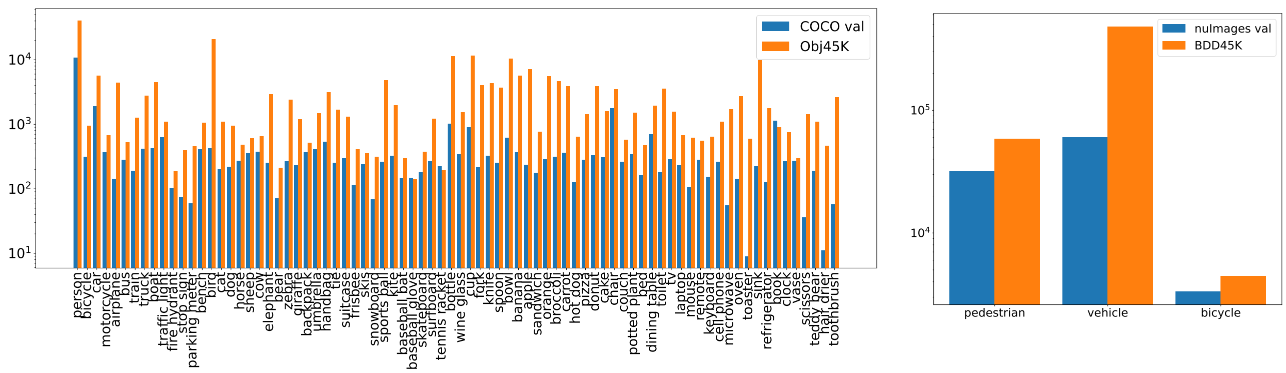
### A.1.2 BDD45K Split

Considering that the widely-used AV datasets [4, 10, 16, 65, 73] have `pedestrian`, `vehicle` and `bicycle` in common, we consider these three classes as ID classes of our SAOD-AV use-case[4]. Then, similar to how we obtain Obj45K, we match these classes of nuImages with the classes of BDD100K, resulting in the following one-to-many matching:

```
’pedestrian’: [’pedestrian’,
”other␣person”],
’vehicle’: [’car’, ’truck’, ’bus’,
’motorcycle’, ’train’, ”trailer”,
”other␣vehicle”],
’bicycle’: ’bicycle’
```

On the other hand, we observe a key difference in annotating `bicycle` and `motorcycle` classes between nuImages and BDD100K datasets. Specifically, while BDD100K has an additional class `rider` that is annotated separately from `bicycle` and `motorcycle` objects, the riders of bicycle and motorcycle are instead included in the annotated bounding box of `bicycle` and `motorcycle` objects in nuImages dataset. In order to align the annotations of these classes between BDD100K and nuImages and provide a consistent evaluation, we aim to rectify the bounding box annotations of these classes in BDD100K dataset such that they follow the annotations of nuImages. Particularly, there

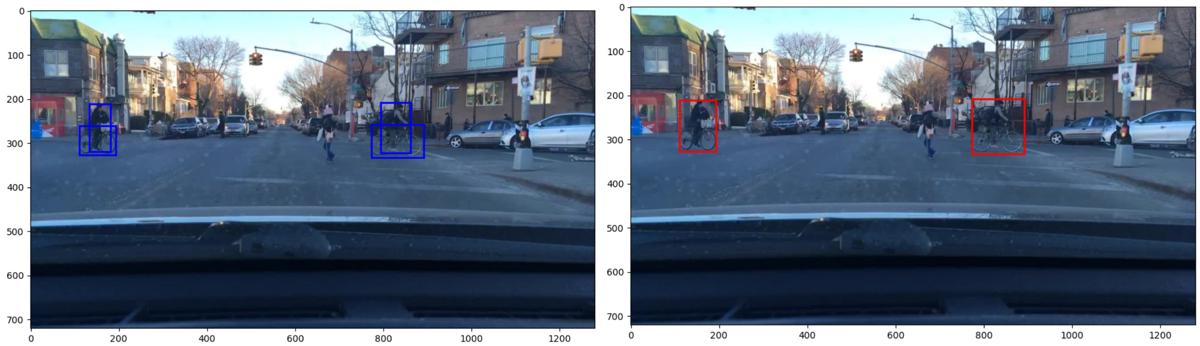
[4] Accordingly, we the models for SAOD-AV for these three classes.



(a) COCO val vs. Obj45K

(b) nuImages val vs. BDD45K

Figure A.6. Distribution of the objects over classes from our test sets and existing val sets. For both SAOD-Gen and SAOD-AV use-cases, our $\mathcal{D}_{\text{ID}}$ have more objects nearly for all classes to provide a thorough evaluation. Note that y-axes are in log-scale.



(a) BDD100K-style annotations

(b) NuImages-style annotations (obtained by Hungarian matching)

Figure A.7. Aligning the annotations of certain classes in BDD100K and nuImages datasets while curating our BDD45K test set. The riders and ridables (bicycles or motorcycles) need to be combined properly in (a). In this example, both of the rider objects are properly assigned to the corresponding bicycle objects by our simple method relying on Hungarian algorithm. In (b), which we use as a test image in our BDD45K, the bounding boxes are combined by finding the smallest enclosing bounding box and the objects are labelled as bicycles.

should be no rider class but `bicycle` and `motorcycle` objects include their riders in the resulting annotations. To do so, we use a simple matching algorithm on BDD100K images to combine bicycle and motorcycle objects with their riders. In particular, given an image, we identify first objects from `bicycle`, `motorcycle` and `rider` categories. Then, we group `bicycle` and `motorcycle` objects as "rideables" and compute IoU between each rideable and rider object. Given this matrix of representing the proximity between each rideable and rider object in terms of their IoUs, we assign riders to rideables by maximizing the total IoU using the Hungarian assignment algorithm [3]. Furthermore, we include a sanity check to avoid possible matching errors, e.g., in which a rideable object might be combined with a rider in a further location in the image due to possible annotation errors. Specifically, our simple sanity is to require a minimum IoU overlap of $0.10$ between a rider and its assigned rideable in the resulting assignment from the Hungarian algorithm. Otherwise, if any of the riders is assigned to a rideable object with an IoU less than $0.10$ in an image, we simply do not include this image in our BDD45K test set. Finally, exploiting the assignment result, we obtain the bounding box annotation using the smallest enclosing bounding box including both the bounding box of the rider and that of the rideable object. As for the category annotation of the object, we simply use the category of the rideable, which is either `bicycle` or `motorcycle`. Fig. A.7 presents an example in which we convert BDD100K annotations of these specific classes into the nuImages format. To validate our approach, we manually examine more than 2500 images in BDD45K test set and observe that it is effective to align the annotations of nuImages and BDD100K.

Overall, using this strategy, we collect 45K images from training and validation sets of BDD100K and construct our BDD45K split. We would like to highlight that our BDD45K dataset is diverse and extensive, where (i) it is

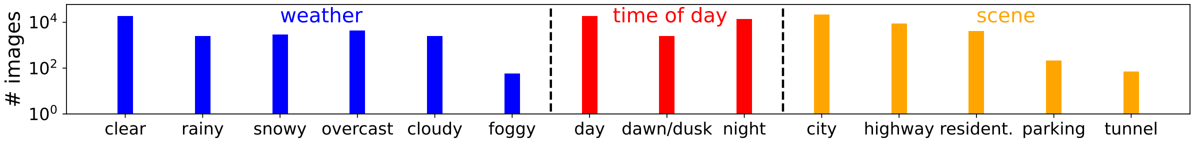


Figure A.8. The diversity of BDD45K split in terms of weather, time of day and scene counts.

larger compared to 16K images of nuImages val set; and (ii) it includes 543K objects in total, significantly larger than the number of objects from these 3 classes in nuImages val set with 96K objects. Please refer to Fig. A.6(b) for quantitative comparison. In terms of diversity, our BDD45K ($\mathcal{D}_{\text{ID}}$) comes from a different distribution than nuImages ($\mathcal{D}_{\text{Train}}$); thereby introducing natural covariate shift. Fig. A.8 illustrates that our BDD45K is very diverse and it is collected from different cities using different camera types than nuImages ($\mathcal{D}_{\text{Train}}$). As a result, as we will see in Sec. B, the accuracy of the models drops significantly from $\mathcal{D}_{\text{Val}}$ to $\mathcal{D}_{\text{ID}}$ even before the corruptions are employed. We note that ImageNet-C corruptions are then applied to this dataset, further increasing the domain shift.

## A.2. Obj45K-C and BDD45K-C Splits

While constructing Obj45K-C and BDD45K-C as $\mathcal{T}(\mathcal{D}_{\text{ID}})$, we use the following 15 different corruptions from 4 main groups [25]:

- *Noise.* gaussian noise, shot noise, impulse noise, speckle noise
- *Blur.* defocus blur, motion blur, gaussian blur
- *Weather.* snow, frost, fog, brightness
- *Digital.* contrast, elastic transform, pixelate, jpeg compression

Then, given an image for a particular severity level that can be 1, 3 or 5, we randomly sample a transformation and apply to the image. In such a way, we obtain 3 different copies of Obj45K and BDD45K during evaluation.

We outline in the definition of the SAOD task (Sec. 3) that an image with a corruption severity 5 might not contain enough cues to perform object detection reliably and that a SAODET is flexible to accept or reject such images as long as it yields accurate and calibrated detections on the accepted ones. To provide insight of providing this flexibility, Fig. A.9 presents example corruptions with severity 5. Note that several cars in the corrupted images above and birds in the ones below are not visible any more due to the severity of the corruption. As notable examples, some of the cars in Fig. A.9(b) and the birds in Fig. A.9(h) Fig. A.9(h) are not visible. As a result, instead of enforcing the detector to predict all of the objects accurately, we do not penalize a detector as long as it can infer that it is uncertain and rejects such images with high corruption severity.

## A.3. SiNObj110K-OOD Split

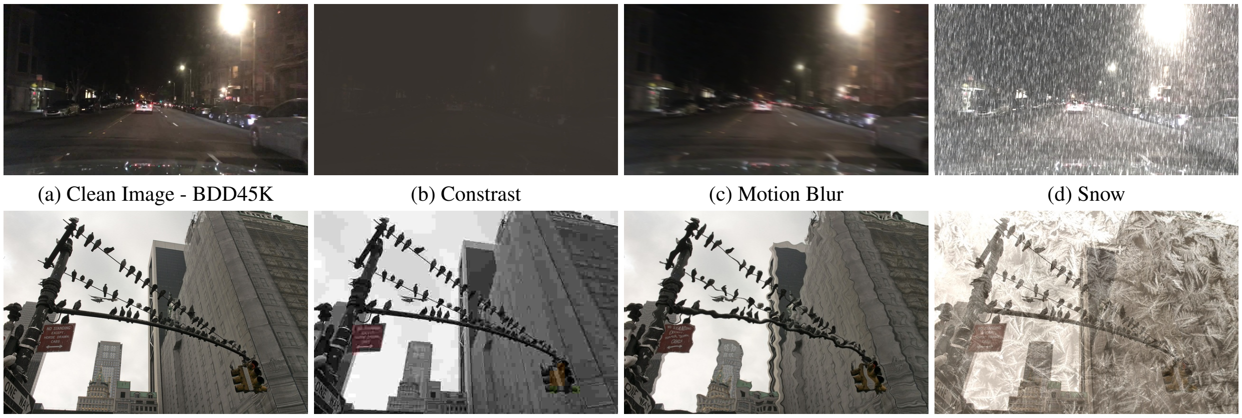
This split is designed to evaluate the reliability of the uncertainties. Following similar work [13, 21], we ensure that the images in our OOD test set do not include any object from ID classes. Specifically, in order to use SiNObj110K-OOD within both SAOD-Gen and SAOD-AV datasets, we select an image to SiNObj110K-OOD if the image does not include an object from either of the ID classes of Obj45K or BDD45K ($\mathcal{D}_{\text{ID}}$). Then, we collect 110K images from three different detection datasets as detailed below:

- *SVHN subset of SiNObj110K-OOD.* We include all 46470 full numbers (not cropped digits) using both training and test sets of SVHN dataset in our OOD test set.
- *iNaturalist OOD subset of SiNObj110K-OOD.* We use the validation set of iNaturalist 2017 object detection dataset to obtain our iNaturalist dataset. Specifically, we include 28768 images in our OOD test set with the following classes:

```
'Actinopterygii', 'Amphibia',
'Animalia', 'Arachnida',
'Insecta', 'Mollusca', 'Reptilia'
```

- *Objects365 OOD subset of SiNObj110K-OOD.* To select images for our OOD test set from Objects365 dataset, we use the following classes as OOD:

```
'Sneakers', 'Other␣Shoes', 'Hat',
'Lamp', 'Glasses', 'Street␣Lights',
'Cabinet/shelf', 'Bracelet',
'Picture/Frame', 'Helmet', 'Gloves',
'Storage␣box', 'Leather␣Shoes', 'Flag',
'Pillow', 'Boots', 'Microphone',
'Necklace', 'Ring', 'Belt',
'Speaker', 'Trash␣bin␣Can', 'Slippers',
'Barrel/bucket', 'Sandals', 'Bakset',
'Drum', 'Pen/Pencil', 'High␣Heels',
'Guitar', 'Carpet', 'Bread', 'Camera',
'Canned', 'Traffic␣cone', 'Cymbal',
'Lifesaver', 'Towel', 'Candle',
'Awning', 'Faucet', 'Tent', 'Mirror',
```



(a) Clean Image - BDD45K (b) Constrast (c) Motion Blur (d) Snow

(e) Clean Image - Obj45K (f) JPEG compression (g) Elastic transform (h) Frost

Figure A.9. Clean and corrupted images using different transformations at severity 5 from AV-OD (upper row) and Gen-OD (lower row) use-cases. We do not penalize a detector if it can infer that it is uncertain and rejects such images with high corruption severity.

```
'Power␣outlet', 'Air␣Conditioner',
'Hockey␣Stick', 'Paddle', 'Ballon',
'Tripod', 'Hanger',
'Blackboard/Whiteboard','Napkin',
'Other␣Fish', 'Toiletry', 'Tomato',
'Lantern', 'Fan', 'Pumpkin',
'Tea␣pot', 'Head␣Phone', 'Scooter',
'Stroller', 'Crane', 'Lemon',
'Surveillance␣Camera', 'Jug', 'Piano',
'Gun', 'Skating␣and␣Skiing␣shoes',
'Gas␣stove', 'Strawberry',
'Other␣Balls', 'Shovel', 'Pepper',
'Computer␣Box', 'Toilet␣Paper',
'Cleaning␣Products', 'Chopsticks',
'Pigeon', 'Cutting/chopping␣Board',
'Marker', 'Ladder', 'Radiator',
'Grape', 'Potato', 'Sausage',
'Violin', 'Egg', 'Fire␣Extinguisher',
'Candy', 'Converter', 'Bathtub',
'Golf␣Club', 'Cucumber',
'Cigar/Cigarette␣', 'Paint␣Brush',
'Pear', 'Hamburger',
'Extention␣Cord', 'Tong', 'Folder',
'earphone', 'Mask', 'Kettle',
'Swing', 'Coffee␣Machine', 'Slide',
'Onion', 'Green␣beans', 'Projector',
'Washing␣Machine/Drying␣Machine',
'Printer', 'Watermelon', 'Saxophone',
'Tissue', 'Ice␣cream', 'Hotair␣ballon',
'Cello', 'French␣Fries', 'Scale',
'Trophy', 'Cabbage', 'Blender',
'Peach', 'Rice', 'Deer', 'Tape',
'Cosmetics', 'Trumpet', 'Pineapple',
'Mango', 'Key', 'Hurdle',
'Fishing␣Rod', 'Medal', 'Flute',
```

```
'Brush', 'Penguin', 'Megaphone',
'Corn', 'Lettuce', 'Garlic',
'Swan', 'Helicopter', 'Green␣Onion',
'Nuts', 'Induction␣Cooker',
'Broom', 'Trombone', 'Plum',
'Goldfish', 'Kiwi␣fruit',
'Router/modem', 'Poker␣Card',
'Shrimp', 'Sushi', 'Cheese',
'Notepaper', 'Cherry', 'Pliers',
'CD', 'Pasta', 'Hammer',
'Cue', 'Avocado', 'Hamimelon',
'Mushroon', 'Screwdriver', 'Soap',
'Recorder', 'Eggplant',
'Board␣Eraser', 'Coconut',
'Tape␣Measur/␣Ruler', 'Pig',
'Showerhead', 'Globe', 'Chips',
'Steak', 'Stapler', 'Campel',
'Pomegranate', 'Dishwasher',
'Crab', 'Meat␣ball', 'Rice␣Cooker',
'Tuba', 'Calculator',
'Papaya', 'Antelope', 'Seal',
'Buttefly', 'Dumbbell',
'Donkey', 'Lion', 'Dolphin',
'Electric␣Drill', 'Jellyfish',
'Treadmill', 'Lighter',
'Grapefruit', 'Game␣board',
'Mop', 'Radish',
'Baozi', 'Target', 'French',
'Spring␣Rolls', 'Monkey', 'Rabbit',
'Pencil␣Case', 'Yak',
'Red␣Cabbage', 'Binoculars',
'Asparagus', 'Barbell',
'Scallop', 'Noddles',
'Comb', 'Dumpling',
'Oyster', 'Green␣Vegetables',
```

Table A.8. COCO-style AP of the used object detectors on validation set ($\mathcal{D}_{\text{Val}}$) and test set ($\mathcal{D}_{\text{ID}}$), along with their corrupted versions ($\mathcal{T}(\mathcal{D}_{\text{Val}})$ and $\mathcal{T}(\mathcal{D}_{\text{ID}})$).

| Dataset | Detector | $\mathcal{D}_{\text{Val}}$ | $\mathcal{T}(\mathcal{D}_{\text{Val}})$ | | | $\mathcal{D}_{\text{ID}}$ | $\mathcal{T}(\mathcal{D}_{\text{ID}})$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | C1 | C3 | C5 | | C1 | C3 | C5 |
| SAOD Gen | F-RCNN | 39.9 | 31.3 | 20.3 | 10.8 | 27.0 | 20.3 | 12.8 | 6.9 |
| | RS-RCNN | 42.0 | 33.7 | 21.8 | 11.6 | 28.6 | 21.7 | 13.7 | 7.3 |
| | ATSS | 42.8 | 33.9 | 22.3 | 11.9 | 28.8 | 22.0 | 14.0 | 7.3 |
| | D-DETR | 44.3 | 36.2 | 24.0 | 12.2 | 30.5 | 23.4 | 15.4 | 8.0 |
| | NLL-RCNN | 40.1 | 31.0 | 20.0 | 11.6 | 26.9 | 20.3 | 12.9 | 6.8 |
| | ES-RCNN | 40.3 | 31.6 | 20.3 | 11.7 | 27.2 | 20.6 | 13.0 | 6.9 |
| SAOD AV | F-RCNN | 55.0 | 44.9 | 31.1 | 16.7 | 23.2 | 19.8 | 12.8 | 7.2 |
| | ATSS | 56.9 | 47.1 | 34.1 | 18.9 | 25.1 | 21.7 | 14.8 | 8.6 |

```
'Cosmetics␣Brush/Eyeliner␣Pencil',
'Chainsaw', 'Eraser', 'Lobster',
'Durian', 'Okra', 'Lipstick',
'Trolley', 'Cosmetics␣Mirror',
'Curling', 'Hoverboard',
'Plate', 'Pot',
'Extractor', 'Table␣Teniis␣paddle'
```

Using both training and validation sets of Objects365, we collect 35190 images that *only* contains objects from above classes.

Consequently, our resulting SiNObj110K-OOD is both diverse and extensive compared to the datasets introduced in previous work [13, 21] which includes around 1-2K images and is collected from a single dataset.

## B. Details of the Used Object Detectors

Here we demonstrate the details of the selected object detectors and ensure that their performance is inline with their expected results. We build our SAOD framework upon the mmdetection framework [7] since it enables us using different datasets and models also with different design choices. As presented in Sec. 3, we use four conventional and two probabilistic object detectors. We exploit all of these detectors for our SAOD-Gen setting by training them on the COCO training set as $\mathcal{D}_{\text{Train}}$. We train all the detectors with the exception of D-DETR. As for D-DETR, we directly employ the trained D-DETR model released in mmdetection framework. This D-DETR model is trained for 50 epochs with a batch size of 32 images on 16 GPUs (2 images/GPU) and corresponds to the vanilla D-DETR (i.e., not its two-stage version and without iterative bounding box refinement).

While training the detectors, we incorporate the multi-scale training data augmentation used by D-DETR into them in order to obtain stronger baselines. Specifically, the multi-scale training data augmentation is sampled randomly from two alternatives: (i) randomly resizing the shorter side of the image within the range of [480, 800] by limiting its longer size to 1333 and keeping the original aspect ratio; or (ii) a sequence of

- randomly resizing the shorter side of the image within the range of [400, 600] by limiting its longer size to 4200 and keeping the original aspect ratio,

- random cropping with a size of [384, 600],

- randomly resizing the shorter side of the cropped image within the range of [480, 800] by limiting its longer size to 1333 and keeping the original aspect ratio.

Unless otherwise noted, we train all of the detectors (as aforementioned, with the exception of D-DETR, which is trained for 50 epochs following its recommended settings [79]) for 36 epochs using 16 images in a batch on 8 GPUs. Following the previous works, we use the initial learning rates of 0.020 for F-RCNN, NLL-RCNN and ES-RCNN; 0.010 for ATSS; and 0.012 for RS-RCNN. We decay the learning rate by a factor of 10 after epochs 27 and 33. As a backbone, we use a ResNet-50 with FPN [41] for all the models, as is common in practice. At test time, we simply rescale the images to 800 × 1333 and do not use any test-time augmentation. For the rest of the design choices, we follow the recommended settings of the detectors.

As for SAOD-AV, we train F-RCNN [61] and ATSS [77] on nuImages training set by following the same design choices. We note that these models are trained using the annotations of the three classes (`pedestrian`, `vehicle` and `bicycle`) in nuImages dataset.

We display baseline results in Tab. A.8 on $\mathcal{D}_{\text{Val}}$, $\mathcal{T}(\mathcal{D}_{\text{Val}})$, $\mathcal{D}_{\text{ID}}$ and $\mathcal{T}(\mathcal{D}_{\text{ID}})$ data splits, which shows the performance on the COCO val set ($\mathcal{D}_{\text{Val}}$ of SAOD-Gen in the table) is inline or higher with those published in the corresponding papers. We would like to note that the performance on $\mathcal{D}_{\text{Val}}$ is lower than that on $\mathcal{D}_{\text{ID}}$ due to (i) more challenging nature of Object365/BDD100K compared to COCO/nuImages and (ii) the domain shift between them. As an example, AP drops $\sim$ 30 points from $\mathcal{D}_{\text{Val}}$ (nuImages) to $\mathcal{D}_{\text{ID}}$ (BDD45K) even before the corruptions are applied. As expected, we also see a decrease in performance with increasing severity of corruptions.

## C. Further Details on Image-level Uncertainty

This section presents further details on image-level uncertainty including the motivation behind; the definitions of the used uncertainty estimation techniques; and more analyses.

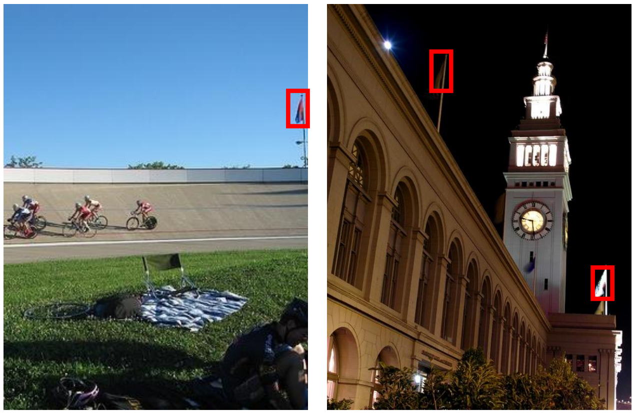


Figure A.10. (Left) An example image from the ID test set and (Right) an example image from the OOD test set used by [13]. The `flag` is not an ID class but exists in both ID and OOD test sets as indicated by red bounding boxes. Labelling the detections corresponding to the `flag` as ID or OOD is non-trivial without labelling every pixel in the images of the training set. Conversely, current works label all detections from an ID image as ID and those from an OOD image as OOD; and then compute AUROC in detection-level using the measured uncertainty. This way of detection-level OOD detection evaluation might not be ideal for object detection.

## C.1. Why is Detection-Level OOD Detection for Object Detection Nontrivial?

As we motivated in Sec. 1 and Sec. 4, evaluating the reliability of uncertainties using OOD detection task in detection-level is conceptually non-trivial for object detection. This is because there is no clear definition as to which detections can be considered ID and which cannot. To elaborate further, unknown objects may appear in two forms at test time: (i) "known-unknowns", which can manifest as background and unlabelled objects in the training set or (ii) "unknown-unknowns", completely unseen objects, which are not present in the training data. It is not possible to split these unknown objects into the two categories without having labels for every pixel in the training set [12]. Current evaluation [13,21] however, does not adhere to this and instead defines "an image" with no ID object as OOD but assumes "any detection" in an OOD image is an OOD detection, and vice versa for an ID image; thereby decreasing the reliability of the evaluation. Fig. A.10 presents an example from an existing ID and OOD test splits [13] to illustrate why the reliability of the evaluation decreases. Conversely, as we have followed, evaluating the reliability of the uncertainties for object detectors based on OOD detection at the image-level aligns with the definition of OOD images, which is again at image-level.

## C.2. Definitions

Here, we provide the definitions of the detection-level uncertainty estimation methods for classification and localisation as well as the aggregation techniques we used to obtain image-level uncertainty estimates.

### C.2.1 Detection-Level Uncertainties

In the following, we present how we obtain detection-level uncertainties from classification and localisation heads. We note that all of these uncertainties, except the uncertainty score, are computed on the raw detections represented by $\{\hat{b}_i^{raw}, \hat{p}_i^{raw}\}^{N^{raw}}$ in Sec. 2 and then propagated through the post-processing steps. The uncertainty score is, instead, directly computed using the confidence of the final detections ($\hat{p}_i$). In such a way, we obtain the uncertainty values of top-$k$ final detections, which are then aggregated for image-level uncertainty estimates.

**Classification Uncertainties** We use the following detection-level classification uncertainties:

- *The entropy of the predictive distribution.* The standard configuration of F-RCNN, NLL-RCNN and ES-RCNN employ a softmax classifier over $K$ ID classes and background; resulting in a $K+1$-dimensional categorical distribution. Denoting this distribution by $\hat{p}_i^{raw}$ (Sec. 2), the entropy of $\hat{p}_i^{raw}$ is:

$$\mathrm{H}(\hat{p}_i^{raw}) = -\sum_{j=1}^{K+1} \hat{p}_{ij}^{raw} \log \hat{p}_{ij}^{raw}, \tag{A.6}$$

  such that $\hat{p}_{ij}^{raw}$ is the probability mass in $j$th class in $\hat{p}_i^{raw}$. As for the object detectors which exploit class-wise sigmoid classifiers, the situation is more complicated since the prediction $\hat{p}_i^{raw}$ comprises of $K$ Bernoulli random variables, instead of a single distribution unlike the softmax classifier. Therefore, we will discuss and analyse different ways of computing entropy for the detectors using sigmoid classifiers in Sec. C.3.1.
- *Dempster-Shafer.* We use the logits as the evidence to compute DS [62]. Accordingly, denoting the $j$th logit (i.e., for class $j$) of the $i$th detection obtained from a softmax-based detector by $s_{ij}$, we compute the uncertainty by

$$\mathrm{DS} = \frac{K+1}{K+1+\sum_{j=1}^{K+1} \exp\left(s_{ij}\right)}, \tag{A.7}$$

  and similarly, for a sigmoid-based classifier yielding $K$ logits, we simply use

$$\mathrm{DS} = \frac{K}{K+\sum_{j=1}^{K} \exp\left(s_{ij}\right)}. \tag{A.8}$$

- *Uncertainty score.* While $\mathrm{H}(\hat{p}_i^{raw})$ and DS are computed on the raw detections, we compute uncertainty score based on final detections using the detection confidence score as $1 - \hat{p}_i$.

**Localisation Uncertainties** We utilise the covariance matrix $\Sigma$ predicted by the probabilistic detectors (NLL-RCNN and ES-RCNN) to compute the uncertainty of a detection in the localisation head. As described in Sec. 3, our models predicts a diagonal covariance matrix,

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}, \tag{A.9}$$

for each detection such that $\sigma_i^2$ with $0 < i \leq 4$ is the predicted variance of the Gaussian for $i$th bounding box parameter. Considering that an increase in $\Sigma$ should imply more uncertainty of the localisation head, we define the following uncertainty measures for localisation exploiting $\Sigma$,.

- *The determinant of the predicted covariance matrix*

$$|\Sigma| = \prod_{i=1}^{4} \sigma_i^2 \tag{A.10}$$

- *The trace of the predicted covariance matrix*

$$\mathrm{tr}(\Sigma) = \sum_{i=1}^{4} \sigma_i^2 \tag{A.11}$$

- *The entropy of the predicted multivariate Gaussian distribution [49]*

$$\mathrm{H}(\Sigma) = 2 + 2\ln(2\pi) + \frac{1}{2}\ln(|\Sigma|) \tag{A.12}$$

### C.2.2 Aggregation Strategies to Obtain Image-Level Uncertainties

In this section, given detection-level uncertainties $\{u_i\}^N$ where $u_i$ is the detection-level uncertainty for the $i$th detection, we present our aggregation strategies to obtain the image-level uncertainty $\mathcal{G}(X)$. Note that $u_i$ corresponds to a detection-level uncertainty after post-processing with top-$k$ survival (Sec. 2), hence implying $N \leq k$. In particular, we use the following aggregation techniques that enables us to obtain reliable image-level uncertainties from different detectors:

- *sum:*

$$\mathcal{G}(X) = \sum_{i=1}^{N} u_i \tag{A.13}$$

Table A.9. AUROC values for different variations of computing entropy as the uncertainty for sigmoid-based detectors. Applying softmax to the logits to obtain $K$-dimensional categorical distribution performs the best for all detectors.

| Detector | average | max class | categorical |
|---|---|---|---|
| RS-RCNN | 73.3 | 91.2 | **93.7** |
| ATSS | 79.9 | 27.5 | **94.3** |
| D-DETR | 63.4 | 27.9 | **93.9** |

- *mean:*

$$\mathcal{G}(X) = \frac{1}{N}\sum_{i=1}^{N} u_i \tag{A.14}$$

- *mean(top-$m$):* Denoting $\phi(i)$ as the index of the $i$th smallest uncertainties,

$$\mathcal{G}(X) = \begin{cases} \frac{1}{m}\sum\limits_{i=1}^{m} u_{\phi(i)}, & \text{if } N \geq m \\ \frac{1}{N}\sum\limits_{i=1}^{N} u_{\phi(i)}, & \text{if } 0 < N < m \end{cases} \tag{A.15}$$

- *min:* Similarly, denoting $u_{\phi(1)}$ is the smallest uncertainty or the most certain one,

$$\mathcal{G}(X) = u_{\phi(1)} \tag{A.16}$$

Finally, we consider the extreme case in which all of the detections are eliminated in the background removal stage, the first step of the post-processing. It is also worth mentioning that this case can be avoided by reducing the score threshold of the detectors, which is typically $0.05$. However, using off-the-shelf detectors by keeping their hyper-parameters as they are, we observe rare cases that a detector may not yield any detection for an image. To give an intuition how rare these cases are, we haven't observed any image with no detection for D-DETR and RS-RCNN and there very few images for F-RCNN. However, for the sake of completeness, we assign a large uncertainty value (typically $10^{12}$) that ensures that the image is classified as OOD in such cases.

## C.3. More Analyses on Image-Level Uncertainty

This section includes more analyses on obtaining image-level uncertainties. We use our Gen-OD setting and report AUROC following Sec. 4 unless explicitly otherwise noted.

### C.3.1 Computing Detection-level Uncertainty for Sigmoid-based Classifiers

Unlike the detectors using $K+1$-class softmax classifiers, the detectors employing sigmoid-based classifiers, such as RS R-CNN, ATSS and D-DETR, yield $K$ different Bernoulli random variables each of which corresponds

Table A.10. Combining classification and localisation uncertainties. Please refer to the text for more details.

| Detector | $\mathrm{H}(\hat{p}_i^{raw})$ | $\mathrm{H}(\Sigma)$ | Balanced | Norm. | AUROC |
|---|---|---|---|---|---|
| NLL-RCNN | ✓ | | | | 92.4 |
| | | ✓ | | | 87.7 |
| | ✓ | ✓ | | | 89.9 |
| | ✓ | ✓ | ✓ | | 92.2 |
| | ✓ | ✓ | ✓ | ✓ | **93.2** |
| ES-RCNN | ✓ | | | | 92.8 |
| | | ✓ | | | 86.4 |
| | ✓ | ✓ | | | 89.3 |
| | ✓ | ✓ | ✓ | | 91.8 |
| | ✓ | ✓ | ✓ | ✓ | **93.2** |

to one of the $K$ different classes in $\mathcal{D}_{\text{Train}}$. In this case, one can think of different ways to compute the detection-level uncertainty. Here, we analyse the effect of three different methods to obtain the uncertainty for a detection as (i) the average over the entropies of $K$ Bernoulli random variables; (ii) the entropy of the maximum-scoring class; and (iii) obtaining a categorical distribution over the classes through softmax first, and then, computing the entropy of this categorical distribution. More precisely, for the $j$th class and the $i$th detection, denoting the predicted logit and corresponding probability (obtained through sigmoid) by $\hat{s}_{ij}$ and $\hat{p}_{ij}^{raw}$ respectively, we define the following uncertainties:

- *The average of the entropies of the $K$ Bernoulli random variables:*

$$\frac{1}{K}\sum_{j=1}^{K}\left(\hat{p}_{ij}^{raw}\log\hat{p}_{ij}^{raw} + (1-\hat{p}_{ij}^{raw})\log(1-\hat{p}_{ij}^{raw})\right); \tag{A.17}$$

- *The entropy of the maximum-scoring class:*

$$\hat{p}_{ik}^{raw}\log\hat{p}_{ik}^{raw} + (1-\hat{p}_{ik}^{raw})\log(1-\hat{p}_{ik}^{raw}) \tag{A.18}$$

  with $k$ being the maximum scoring class for detection $i$; and
- *The entropy of the categorical distribution:* It is simply obtained by first applying softmax over $K$-dimensional logits, yielding a categorical distribution, and then, computing the entropy following Eq. (A.6) with $K$ classes.

Tab. A.9 indicates that the entropy of the categorical distribution performs consistently the best for all three sigmoid-based detectors.

### C.3.2 Combining Classification and Localisation Uncertainties

Considering that the probabilistic object detectors yield an uncertainty both for classification and localisation for each detection, here we investigate whether there is a benefit in combining such uncertainties for a detection. Basically, we combine the entropy of the predictive classification distribution ($\mathrm{H}(\hat{p}_i^{raw})$) and the entropy of the predictive Gaussian distribution for localisation ($\mathrm{H}(\Sigma)$). Assuming these two distributions independent, the entropy of the joint distribution can be obtained by the summation over the entropies, i.e., $\mathrm{H}(\hat{p}_i^{raw}) + \mathrm{H}(\Sigma)$. However, this way of combining tends to overestimate the contribution of the localisation as the localisation output involves four random variables but the classification is univariate. Consequently, we first increase the contribution of the classification by multiplying its entropy by 4, which results in a positive effect ("balanced" in Table A.10). We also find it useful to normalize the uncertainties between $0$ and $1$ using their minimum and maximum scores obtained on the validation set. This normalisation shows for the both cases that the resulting performance is better compared to using only classification and localisation ("Norm." in Table A.10).

We also would like to note that the resulting AUROC does not outperform using only the uncertainty score $(1-\hat{p}_i)$, which yields $94.1$ AUROC score with mean(top-3) for both of the detectors ES-RCNN and NLL-RCNN as demonstrated in Tab. 2. Hence, instead of combining classification and localisation uncertainties, we still suggest using $1-\hat{p}_i$ to obtain the detection-level uncertainties.

### C.3.3 The Effect of Aggregation Techniques on a Localisation Uncertainty Estimate

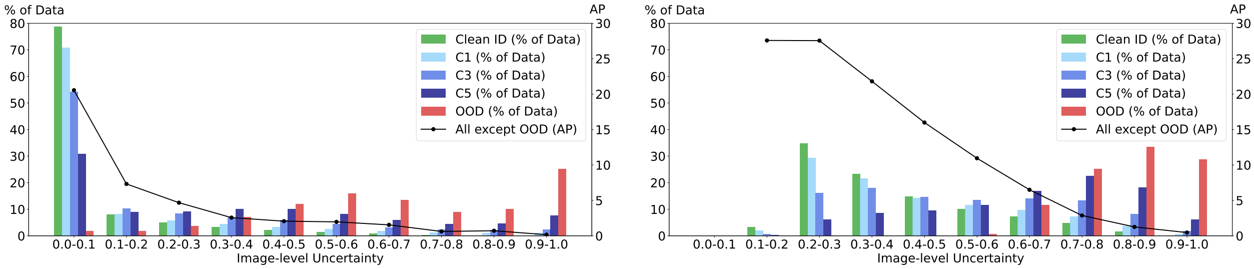
In Tab. 2, we investigated the aggregation methods using the uncertainty score, which is a classification-based uncertainty estimate. Here in Tab. A.11, we present an extended version of Tab. 2 including the effect of the same aggregation methods on $|\Sigma|$ as one example of our localisation uncertainty estimation methods. Tab. A.11 also validates our conclusion on $|\Sigma|$ to use the most certain detections in obtaining image-level uncertainties.

### C.3.4 On the Reliability of Image-level Uncertainties

Here, similar to our analysis on SAOD-Gen, we present the distribution of the image-level uncertainties for different subsets of $\mathcal{D}_{\text{Test}}$ on SAOD-AV. Fig. A.11 confirms that the following observations obtained on SAOD-Gen are also valid for SAOD-AV: (i) the distribution gets more right-skewed as the subset moves away from the $\mathcal{D}_{\text{ID}}$ and (ii) AP (black line) perfectly decreases as the uncertainties increase (refer to Sec. 4 for details). These figures confirm on a different dataset that the image-level uncertainties are reliable and effective.

Table A.11. AUROC scores of different aggregations to obtain image-level uncertainty. $|\Sigma|$ can be computed for probabilistic detectors, hence N/A for others. mean(top-m) refers to average of the uncertainties of the most $m$ certain detections (the detections with the least uncertainty based on $1 - \hat{p}_i$ or $|\Sigma|$). Using few most certain detections perform better for both detection-level uncertainty estimation methods. Underlined & Bold: best of a detector, bold: second best.

| Dataset ($\mathcal{D}_{\text{ID}}$ vs. $\mathcal{D}_{\text{OOD}}$) | Detector | sum | | mean | | mean(top-5) | | mean(top-3) | | mean(top-2) | | min | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $1 - \hat{p}_i$ | $\|\Sigma\|$ | $1 - \hat{p}_i$ | $\|\Sigma\|$ | $1 - \hat{p}_i$ | $\|\Sigma\|$ | $1 - \hat{p}_i$ | $\|\Sigma\|$ | $1 - \hat{p}_i$ | $\|\Sigma\|$ | $1 - \hat{p}_i$ | $\|\Sigma\|$ |
| SAOD-Gen | F R-CNN [61] | 20.9 | N/A | 84.1 | N/A | 93.4 | N/A | **94.1** | N/A | $\underline{\textbf{94.4}}$ | N/A | 93.8 | N/A |
| | RS R-CNN [55] | 85.8 | N/A | 85.8 | N/A | 94.3 | N/A | $\underline{\textbf{94.8}}$ | N/A | $\underline{\textbf{94.8}}$ | N/A | 93.5 | N/A |
| | ATSS [77] | 66.2 | N/A | 86.3 | N/A | 93.8 | N/A | $\underline{\textbf{94.2}}$ | N/A | **94.0** | N/A | 92.6 | N/A |
| | D-DETR [79] | 85.2 | N/A | 85.2 | N/A | 94.4 | N/A | **94.7** | N/A | 94.6 | N/A | 93.3 | N/A |
| | NLL R-CNN [23] | 22.6 | 41.6 | 83.8 | 74.9 | 93.4 | 87.4 | **94.1** | 87.6 | $\underline{\textbf{94.4}}$ | 87.5 | 93.7 | 87.0 |
| | ES R-CNN [21] | 22.1 | 24.5 | 84.6 | 32.9 | 93.4 | 83.8 | **94.1** | 85.0 | $\underline{\textbf{94.4}}$ | 85.7 | 93.8 | 86.3 |
| SAOD-AV | F R-CNN [61] | 27.1 | N/A | 84.1 | N/A | 96.4 | N/A | **97.3** | N/A | $\underline{\textbf{97.4}}$ | N/A | 96.0 | N/A |
| | ATSS [77] | 18.8 | N/A | 92.2 | N/A | $\underline{\textbf{97.7}}$ | N/A | **97.6** | N/A | 97.3 | N/A | 95.7 | N/A |



(a) F-RCNN

(b) ATSS

Figure A.11. The distribution of the image-level uncertainties obtained from different detectors on clean ID, corrupted ID with severities 1, 3, 5 and OOD data on SAOD-AV dataset.

Table A.12. Effectiveness of our pseudo-OOD set approach compared to using TPR@0.95.

| Task | Detector | Method | BA | TPR | TNR |
|---|---|---|---|---|---|
| Gen-OD | F-RCNN | TPR@0.95 | 83.2 | 98.5 | 72.0 |
| | | pseudo-OOD | **87.7** | 94.7 | 81.6 |
| | RS-RCNN | TPR@0.95 | 84.0 | 98.3 | 73.4 |
| | | pseudo-OOD | **88.9** | 92.8 | 85.3 |
| | ATSS | TPR@0.95 | 84.7 | 96.9 | 75.2 |
| | | pseudo-OOD | **87.8** | 93.1 | 83.0 |
| | D-DETR | TPR@0.95 | 85.8 | 97.2 | 76.8 |
| | | pseudo-OOD | **88.9** | 90.0 | 87.8 |
| SAOD-AV | F-RCNN | TPR@0.95 | 80.9 | 97.7 | 69.1 |
| | | pseudo-OOD | **91.0** | 94.1 | 88.2 |
| | ATSS | TPR@0.95 | 83.5 | 96.7 | 73.5 |
| | | pseudo-OOD | **85.8** | 95.9 | 77.6 |

#### C.3.5 The Effectiveness of Using Pseudo OOD val set for Image-level Uncertainty Thresholding

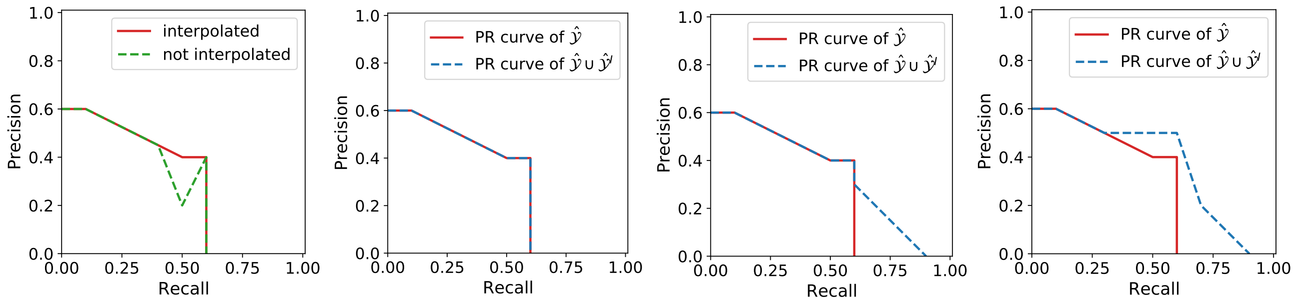
In order to compute the image-level uncertainty threshold $\bar{u}$ and decide whether or not to accept an image, we presented a way to construct pseudo-OOD val set in Sec. 4 as $\mathcal{D}_{\text{Val}}$ only includes ID images. Here, we discuss the effectiveness of this pseudo-set approach. To do so, we also prefer to have a baseline to compare our method against and demonstrate its effectiveness. However, to the best of our knowledge, there is no such a method that obtains a threshold relying only on the ID data for OOD detection task. As a result, inspired from the performance measure TPR@0.95 [13], we simply set the threshold $\bar{u}$ to the value that corresponds to TPR@0.95, and use it as a baseline. Note that this approach only relies on the ID val set= and hence there is no need for OOD val set, which is similar to our pseudo-OOD approach. Tab. A.12 compares our pseudo-OOD approach with TPR@0.95 baseline; suggesting, on average, more than 4.5 BA gain over the baseline method; thereby confirming the effectiveness of our approach.

## D. Further Details on Calibration of Object Detectors

This section provides further details and analyses on calibration of object detectors.

### D.1. How does AP benefit from low-scoring detections?

Here we show that enlarging the detection set with low-scoring detections provably does not decrease AP. Thereby confirming the practical limitations previously discussed by



(a) Interpolating PR Curve (b) Case 1 of the proof (c) Case 2 of the proof (d) Case 3 of the proof

Figure A.12. Illustrations of **(a)** non-interpolated and interpolated PR curves. Typically, the area under the interpolated PR curve is used as the AP value in object detection; **(b), (c), (d)** corresponding to each of the three different cases we consider in the proof of Theorem 1. Following Theorem 1, in all three cases, the area under the red curve is smaller or equal to that of the blue curve.

Oksuz et al. [58]. As a result, instead of top-k predictions and AP, we require a thresholded detection set in SAOD task and employ the LRP Error as a measure of accuracy to enforce this type of output.

Before proceeding, below we provide a formal definition of AP as a basis to our proof.

**Definition of AP.** AP is defined as the area under the Precision-Recall curve [15, 53, 58]. Here we formalize how to obtain this curve and the resulting AP in object detection given detections and the ground truths. Considering the common practice, we will first focus on the AP of a single class and then discuss further after our proof. More precisely, computing AP for class $c$ from an IoU threshold of $\tau$, two sets are required:

- *A set of detections obtained on the test set:* This set is represented by tuples $\hat{\mathcal{Y}} = \{\hat{b}_i, \hat{p}_i, X_i\}^{N_c}$, where $\hat{b}_i$ and $\hat{p}_i$ are the bounding box and confidence score of the $i$th detection respectively. $X_i$ is the image id that the $i$th detection resides and $N_c$ is the number of all detections across the dataset from class $c$. We assume that the detections obtained from a single image is less than $k$ where $k$ represents the upper-bound within the context of the top-$k$ survival (Sec. 2), that is, there can be up to $k$ detections from each image.
- *A set of ground truths of the test set:* This set is represented by tuples $\mathcal{Y} = \{b_i, X_i\}^{M_c}$, where $b_i$ is the bounding box of the ground truth and $X_i$ is similarly the image id. $M_c$ is the number of total ground truth objects from class $c$ across the dataset.

Then, the detections are matched with the ground truths to identify TP and FP detections using a matching algorithm. For the sake of completeness, we provide a matching algorithm that is used by the commonly-used COCO benchmark [43]:

1. All detections in $\hat{\mathcal{Y}}$ are first sorted with respect to the confidence score in descending order
2. Then, going over this sorted list of detections, the $j$th detection is identified as a TP if there exists a ground truth that satisfies the following two conditions:
   - The ground truth is not previously assigned to any other detections with a larger confidence score than that of $j$,
   - The IoU between the ground truth and $j$th detection is more than $\tau$, the TP validation threshold.

   Note that the second condition also implies that the $j$th detection and the ground truth that it matches with should reside in the same image. If there is a single ground truth satisfying these two conditions, then $j$ is matched with that ground truth; else if there are more than one ground truths ensuring these conditions, then the $j$th detection is matched with the ground truth that $j$ has the largest IoU with.

3. Upon completing this sorted list, the detections that are not matched with any ground truths are identified as FPs.

This matching procedure enables us to determine which detections are TP or FP. Now, let $L = [L_1, ..., L_{N_c}]$ be a binary vector that represents whether $j$th detection is a TP or FP and assume that $L$ is also sorted with respect to the confidence scores of the detections. Specifically, $L_i = 1$ if the $i$th detection is a TP, else the $i$th detection is FP and $L_i = 0$. Consequently, we need precision and recall pairs in order to obtain the Precision-Recall curve, area under which corresponds to the AP. Noting that the precision is the ratio between the number of TPs and number of all detections; and recall is the ratio between the number of TPs and number of ground truths, we can obtain these pairs by leveraging $L$. Denoting the precision and recall vectors by $Pr = [Pr_1, ..., Pr_{N_c}]$ and $Re = [Re_1, ..., Re_{N_c}]$ respectively, the $i$th element of these vectors can be obtained by:

$$Pr_i = \frac{\sum_{k=1}^{i} L_k}{i}, \text{ and } Re_i = \frac{\sum_{k=1}^{i} L_k}{M_c}. \tag{A.19}$$

Since these obtained precision values $Pr_i$ may not be monotonically decreasing function of recall, there can be wiggles in the Precision-Recall curve. Therefore, it is common in object detection [15, 18, 43] to interpolate the precisions $Pr$ to make it monotonically decreasing with respect to the recall $Re$. Denoting the interpolated precision vector by $\bar{Pr} = [\bar{Pr}_1, ..., \bar{Pr}_{N_c}]$, its $i$th element $\bar{Pr}_i$ is obtained as follows:

$$\bar{Pr}_i = \max_{i:Re_i \geq Re_k} (Pr_i). \tag{A.20}$$

Finally, Eq. (A.20) also allows us to interpolate the PR curve to the precision and recall axes. Namely, we include the pairs that (i) $\bar{Pr}_1$ with recall 0; and (ii) precision 0 with recall $\bar{Re}_{N_c}$. This allows us to obtain the final Precision-Recall curve using these two additional points as well as the vectors $\bar{Pr}_i$ and $Re_i$. Then, the area under this curve corresponds to the Average Precision of the detection set $\hat{\mathcal{Y}}$ for the IoU validation threshold of $\tau$, which we denote as $\mathrm{AP}_\tau(\hat{\mathcal{Y}})$. As an example, Fig. A.12(a) illustrates a PR curve before and after interpolation. Based on this definition, we now prove that low-scoring detections do not harm AP.

**Theorem 1.** *Given two sets of detections* $\hat{\mathcal{Y}} = \{\hat{b}_i, \hat{p}_i, X_i\}_{i=1}^{N_c}$, $\hat{\mathcal{Y}}' = \{\hat{b}_j, \hat{p}_j, X_j\}_{j=1}^{N_c^{'}}$ *and denoting* $p_{min} = \min_{\{\hat{b}_i, \hat{p}_i, X_i\} \in \hat{\mathcal{Y}}} \hat{p}_i$, $p^{'}_{max} = \max_{\{\hat{b}_j, \hat{p}_j, X_j\} \in \hat{\mathcal{Y}}'} \hat{p}_j$, *if* $p^{'}_{max} < p_{min}$, *then* $\mathrm{AP}_\tau(\hat{\mathcal{Y}}) \leq \mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$.

*Proof.* We denote the recall and precision values to compute $\mathrm{AP}_\tau(\hat{\mathcal{Y}})$ by $Pr = [Pr_1, ..., Pr_{N_c}]$ and $Re = [Re_1, ..., Re_{N_c}]$, and similarly the interpolated precision is $\bar{Pr} = [\bar{Pr}_1, ..., \bar{Pr}_{N_c}]$. We aim to obtain these vectors for $\mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ to be able to compare the resulting $\mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ with $\mathrm{AP}_\tau(\hat{\mathcal{Y}})$. To do so, we introduce $Pr^{'}$, $Re^{'}$ and $\bar{Pr}^{'}$ as the precision, recall and the interpolated precision vectors of $\mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ respectively.

By definition, the numbers of elements in $Pr^{'}$, $Re^{'}$ and $\bar{Pr}^{'}$ are equal to the number of detections in $\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'}$, which is simply $N_c + N_c^{'}$. More precisely, we need to determine the following three vectors to be able to obtain $\mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$:

$$Pr^{'} = \{Pr^{'}_1, ..., Pr^{'}_{N_c}, Pr^{'}_{N_c+1}, ..., Pr^{'}_{N_c+N_c^{'}}\} \tag{A.21}$$

$$Re^{'} = \{Re^{'}_1, ..., Re^{'}_{N_c}, Re^{'}_{N_c+1}, ..., Re^{'}_{N_c+N_c^{'}}\} \tag{A.22}$$

$$\bar{Pr}^{'} = \{\bar{Pr}^{'}_1, ..., \bar{Pr}^{'}_{N_c}, \bar{Pr}^{'}_{N_c+1}, ..., \bar{Pr}^{'}_{N_c+N_c^{'}}\} \tag{A.23}$$

As an additional insight to those three vectors, $p^{'}_{max} < p_{min}$ implies the following:

- The first $N_c$ elements of $Pr^{'}$, $Re^{'}$ and $\bar{Pr}^{'}$ account for the precision, recall and interpolated precision values computed on the detection from $\hat{\mathcal{Y}}$; and
- their elements between $N_c$+1 to the last element ($N_c$+ $N_c^{'}$) correspond to the precision, recall and interpolated precision values computed on the detections from $\hat{\mathcal{Y}}^{'}$.

Note that by definition, computing precision and recall on the $i$th detection only considers the detections with higher scores than that of $i$ (and ignores the ones with lower scores than that of $i$), since the list of labels denoted by $L$ in Eq. (A.19) is sorted with respect to the confidence scores. As a result, the following holds for precision and recall values (but not the interpolated precision):

$$Pr^{'}_i = Pr_i, \text{ and } Re^{'}_i = Re_i \text{ for } i \leq N_c. \tag{A.24}$$

Then, the difference between $\mathrm{AP}_\tau(\hat{\mathcal{Y}})$ and $\mathrm{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ depends on two aspects:

1. $Pr^{'}_i$ and $Re^{'}_i$ for $N_c < i \leq N_c + N_c^{'}$
2. the interpolated precision vector $\bar{Pr}^{'}$ of $\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'}$, to be obtained using $Pr^{'}$ and $Re^{'}$ based on Eq. (A.20)

For the rest of the proof, we enumerate all possible three cases for $\hat{\mathcal{Y}}^{'}$ and identify these aspects.

**Case (1): $\hat{\mathcal{Y}}^{'}$ does not include any TP.** This case suggests that the detections in $\hat{\mathcal{Y}}^{'}$ are all FPs, and neither the number of TPs nor the number of FNs change for $N_c < i \leq N_c + N_c^{'}$, implying:

$$Re^{'}_i = Re^{'}_{N_c}, \text{ for } N_c < i \leq N_c + N_c^{'}. \tag{A.25}$$

As for the precision, it is monotonically decreasing as $i$ increases between $N_c < i \leq N_c + N_c^{'}$ since the number of FPs increases, that is,

$$Pr^{'}_{i-1} > Pr^{'}_i, \text{ for } N_c < i \leq N_c + N_c^{'}. \tag{A.26}$$

Having identified $Pr^{'}_i$ and $Re^{'}_i$ for $N_c < i \leq N_c + N_c^{'}$, now let's obtain the interpolated precision $\bar{Pr}^{'}$. To do so, we focus on $\bar{Pr}^{'}$ in two parts: Up to and including its $N_c$th element and its remaining part. Since $Pr^{'}_{N_c} > Pr^{'}_i$, for $N_c < i \leq N_c + N_c^{'}$, the low-scoring detections in $\hat{\mathcal{Y}}^{'}$ do not affect $\bar{Pr}^{'}_i$ for $i \leq N_c$ considering Eq. (A.20), implying:

$$\bar{Pr}^{'}_i = \bar{Pr}_i, \text{ for } i \leq N_c. \tag{A.27}$$

As for $N_c < i \leq N_c + N_c^{'}$, since $Re_k = Re_i$, $\bar{Pr}^{'}_i = \bar{Pr}^{'}_{N_c}$ holds.

As a result, the detections from $\hat{\mathcal{Y}}^{'}$ will have all equal recall and interpolated precision, which is also equal to $\bar{Pr}_{N_c}$ and $Re_{N_c}$; implying that they do not introduce new points to the Precision-Recall curve used to obtain $\text{AP}_\tau(\hat{\mathcal{Y}})$. Therefore, $\text{AP}_\tau(\hat{\mathcal{Y}}) = \text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ in this case.

Fig. A.12(b) illustrates this case to provide more insight. In particular, when there is no TP in the low-scoring detections ($\hat{\mathcal{Y}}^{'}$), then no new points are introduced compared to the PR curve of $\hat{\mathcal{Y}}$ and the resulting AP after including low-scoring detections does not change.

**Case (2): $\hat{\mathcal{Y}}^{'}$ includes TPs and $\max\limits_{N_c<i\leq N_c+N_c^{'}}(Pr_i^{'}) \leq \min\limits_{i\leq N_c}(\bar{Pr}_i)$.** Note that $\max\limits_{N_c<i\leq N_c+N_c^{'}}(Pr_i^{'}) \leq \min\limits_{i\leq N_c}(\bar{Pr}_i)$ implies that the interpolated precisions computed on the detection set $\hat{\mathcal{Y}}$ ($\bar{Pr}_i^{'}$ for $i \leq N_c$) will not be affected by the detections in $\hat{\mathcal{Y}}^{'}$. As a result, Eq. (A.24) can simply be extended to the interpolated precisions:

$$\bar{Pr}_i^{'} = \bar{Pr}_i \text{ for } i \leq N_c. \tag{A.28}$$

Considering the area under the curve of the pairs $\bar{Pr}_i^{'}$ and $Re_i^{'} = Re_i$ for $i \leq N_c$, it is already guaranteed that $\text{AP}_\tau(\hat{\mathcal{Y}}) \leq \text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ completing the proof for this case.

To provide more insight, we also briefly explore the effect of remaining detections, that are the detections in $N_c < i \leq N_c + N_c^{'}$ and include TPs. Assume that $j$th detection is the TP with the highest confidence score within the detections for $N_c < i \leq N_c + N_c^{'}$. Then, for the $j$th detection $0 < \bar{Pr}_j^{'} < \bar{Pr}_{N_c}^{'}$ as $\max\limits_{N_c<i\leq N_c+N_c^{'}}(Pr_i^{'}) \leq \min\limits_{i\leq N_c}(\bar{Pr}_i)$ by definition. Moreover, since the number of TPs increases and the number of ground truths is a fixed number, $Re_j^{'} > Re_{N_c}^{'}$. This implies that, the PR curve now has $\bar{Pr}_j^{'} > 0$ precision for some $Re_j^{'}$. Note that the precision was implicitly $0$ for $Re_j^{'}$ for the detection set $\hat{\mathcal{Y}}$ since this new ground truth could not be retrieved regardless of the number of predictions. Accordingly, the additional area under the PR curve of $\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'}$ compared to that of $\hat{\mathcal{Y}}$ increases and it is guaranteed that $\text{AP}_\tau(\hat{\mathcal{Y}}) < \text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ in this case. As depicted in Fig. A.12(c), the area-under-the PR curve of $\hat{\mathcal{Y}}$ is extended towards higher recall (compare blue curve with the red one) resulting in a larger $\text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$ compared to $\text{AP}_\tau(\hat{\mathcal{Y}})$.

**Case (3): $\hat{\mathcal{Y}}^{'}$ includes TPs and $\max\limits_{N_c<i\leq N_c+N_c^{'}}(Pr_i^{'}) > \min\limits_{i\leq N_c}(\bar{Pr}_i)$.** Unlike case (ii), this case implies that upon merging $\hat{\mathcal{Y}}$ and $\hat{\mathcal{Y}}^{'}$, some of the $\bar{Pr}_i$ of $\hat{\mathcal{Y}}$ with $Pr_j^{'} > \bar{Pr}_i$ will be replaced by a larger value due to Eq. (A.20), i.e., $\bar{Pr}_i^{'} > \bar{Pr}_i$ for some $i$ while the rest will be equal similar to Case (ii). This simply implies that $\text{AP}_\tau(\hat{\mathcal{Y}}) < \text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$.

Fig. A.12(d) includes an illustration for this case demonstrating that the PR curve of $\hat{\mathcal{Y}}$ is extended in both of the axes: (i) owing to the interpolation thanks to a TP in $\hat{\mathcal{Y}}^{'}$ with higher precision in $\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'}$, it is extended in precision axis; and (ii) thanks to a new TP in $\hat{\mathcal{Y}}^{'}$, it is extended in recall axis. Note that in our proof for this case, we only discussed the extension in precision since each of the extensions is sufficient to show $\text{AP}_\tau(\hat{\mathcal{Y}}) < \text{AP}_\tau(\hat{\mathcal{Y}} \cup \hat{\mathcal{Y}}^{'})$. □

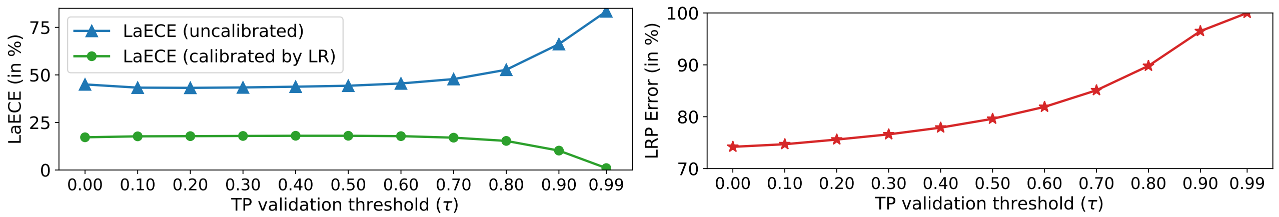
**Discussion** Theorem 1 can also be extended to COCO-style AP. To be more specific and revisit the definition of COCO-style AP, first the class-wise COCO-style APs are obtained by averaging over the APs computed over $\tau \in \{0.50, 0.55, ..., 0.95\}$ for a single class. Then, the detector COCO-style AP is again the average of the class-wise APs. Considering that the arithmetic mean is a monotonically increasing function, Theorem 1 also applies to the class-wise COCO-style AP and the detector COCO-style AP. More precisely, in the case that either Case (1) applies for some (or all) of the classes and the detections for the remaining classes stay the same, then following Case (1), COCO-style AP does not change. That is also the reason why we do not observe a change in COCO-Style AP in Fig. 5(a) once we add dummy detections that are basically FPs with lower scores. If at least for a single class Case (2) or (3) apply, then COCO-style AP increases considering the monotonically increasing nature of the arithmetic average. Following from this, we observe some decrease in COCO-style AP when we remove the detections in Fig. 5(b) when we threshold and remove some TPs. As a result, we conclude that AP, including COCO-Style AP, encourages the detections with lower scores.

## D.2. Sensitivity of LAECE to TP validation threshold

Here we analyse the sensitivity of LAECE to the TP validation threshold $\tau$. Please note that we normally obtain class-wise LRP-optimal thresholds $\bar{v}$ considering a specific $\tau$ on $\mathcal{D}_{\text{Val}}$, then use the resulting detections while measuring the LRP Error and LAECE on the test set using the same IoU validation threshold $\tau$. Namely, we use $\tau$ for two purposes: (i) to obtain the thresholds $\bar{v}$; and (ii) to evaluate the resulting detections in terms of LAECE and LRP Error. As we aim to understand how LAECE, as a performance measure, behaves under different specifications of the TP validation threshold $\tau$ here, we decouple these two purposes of $\tau$ by fixing the detection confidence threshold $\bar{v}$ to the value obtained from a TP validation threshold of $0.10$. This enables us to fix the detection set as the input of LAECE and only focus on how the performance measures behave when only $\tau$ changes.

Specifically, we use F-RCNN detector, validate $\bar{v}$ on COCO validation set and obtain the detections on Obj45K test set using $\bar{v}$. Then, given this set of detections, for different values of $\tau \in [0, 1]$, we compute:

- LAECE for uncalibrated detection confidence scores;



(a) Sensitivity of LaECE to $\tau$

(b) Sensitivity of LRP Error to $\tau$

Figure A.13. Sensitivity analysis of LaECE and LRP Error. We use the detections of F-RCNN on our Obj45K split. **(a)** For both calibrated and uncalibrated case, we observe that the LaECE is not sensitive for $\tau \in [0.0, 0.5]$. When $\tau$ gets larger, the misalignment between detection scores and performance increases for the uncalibrated case, while the calibration becomes an easier problem since most of the detections are now FP. In the extreme case that $\tau = 1$ (a perfect localisation is required for a TP), there is no TP and it is sufficient to assign a confidence score of 0.00 to all of the detections to obtain 0 LaECE. **(b)** Sensitivity analysis of LRP Error. As also previously analysed [58], when $\tau$ increases, number of FPs increases and LRP increases. In the extreme case when $\tau \approx 1$, LRP approximates 1.

- LaECE for calibrated detection confidence scores using linear regression (LR); and

- LRP Error.

Fig. A.13 demonstrates how these three quantities change for $\tau \in [0, 1]$. Note that both for the uncalibrated and calibrated cases, LaECE is not sensitive for $\tau \in [0.0, 0.5]$. As for $\tau \in [0.5, 1.0]$, LaECE increases for the uncalibrated case due to the fact that the detection task becomes more challenging once a larger TP validation threshold $\tau$ is required and that the uncalibrated detections implies more over-confidence as $\tau$ increases. Conversely, in this case, the calibration task becomes easier as the most of the detections are now FP. As an insight, please consider the the extreme case that $\tau = 1$ in which a perfect localisation is required for a TP. In this case, there is no TP and it is sufficient for a calibrator to assign a confidence score of 0.00 to all detections and achieve perfect LaECE that is 0. Finally, as also analysed before [58], when $\tau$ increases, the detection task becomes more challenging, and therefore LRP Error, as the lower-better measure of accuracy, also increases. This is because the number of TPs decreases and the number of FPs increases as $\tau$ increases.

While choosing the TP validation threshold $\tau$ for our SAOD framework, we first consider that a proper $\tau$ should decompose the false positive and localisation errors properly. Having looked at the literature, the general consensus of object detection analysis tools [1, 26] to split the false positive and localisation errors is achieved by employing an IoU of 0.10. As a result, following these works, we set $\tau = 0.10$ throughout the paper unless otherwise noted. Still, the TP validation threshold $\tau$ should be chosen by the requirements of the specific application.

### D.3. Derivation of Eq. (5)

In Sec. 5.3, we claim that the LaECE for a bin reduces to:

$$\left| \sum_{\substack{\hat{b}_i \in \hat{\mathcal{D}}^c_j \\ \psi(i)>0}} \left( t_i^{cal} - \mathrm{IoU}(\hat{b}_i, b_{\psi(i)}) \right) + \sum_{\substack{\hat{b}_i \in \hat{\mathcal{D}}^c_j \\ \psi(i)\leq 0}} t_i^{cal} \right|, \tag{A.29}$$

which allows us to set the target $t_i^{cal}$ as:

$$t_i^{cal} = \begin{cases} \mathrm{IoU}(\hat{b}_i, b_{\psi(i)}), & \text{if } b_{\psi(i)} > 0 \text{ ($i$ is true positive),} \\ 0, & \text{otherwise ($i$ is false positive).} \end{cases} \tag{A.30}$$

In this section, we derive (A.29) given the definition of LaECE in Eq. (4) to justify our claim.

To start with, Eq. (4) defines LaECE for class $c$ as

$$\mathrm{LaECE}^c = \sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}^c_j|}{|\hat{\mathcal{D}}^c|} \left| \bar{p}^c_j - \mathrm{precision}^c(j) \times \bar{\mathrm{IoU}}^c(j) \right|, \tag{A.31}$$

which can be expressed as,

$$\sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}^c_j|}{|\hat{\mathcal{D}}^c|} \left| \bar{p}_j - \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} 1}{|\hat{\mathcal{D}}^c_j|} \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} 1} \right|, \tag{A.32}$$

as

$$\mathrm{precision}^c(j) = \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} 1}{|\hat{\mathcal{D}}^c_j|}, \tag{A.33}$$

and,

$$\bar{\mathrm{IoU}}^c(j) = \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} 1}. \tag{A.34}$$

The expression $\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} 1$ (in the nominator of $\mathrm{precision}^c(j)$ and the denominator of $\bar{\mathrm{IoU}}^c(j)$) corresponds to the number of TPs. Canceling out these terms yield

$$\sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}^c_j|}{|\hat{\mathcal{D}}^c|} \left| \bar{p}_j - \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{|\hat{\mathcal{D}}^c_j|} \right|. \quad \text{(A.35)}$$

$\bar{p}_j$, the average of the confidence score in bin $j$, can similarly be obtained as:

$$\bar{p}_j = \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j} \hat{p}_k}{|\hat{\mathcal{D}}^c_j|}, \quad \text{(A.36)}$$

and replacing $\bar{p}_j$ in Eq. (A.35) yields

$$\sum_{j=1}^{J} \frac{|\hat{\mathcal{D}}^c_j|}{|\hat{\mathcal{D}}^c|} \left| \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j} \hat{p}_k}{|\hat{\mathcal{D}}^c_j|} - \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{|\hat{\mathcal{D}}^c_j|} \right|. \quad \text{(A.37)}$$

Since $a|x| = |ax|$ if $a \geq 0$, we take $\frac{|\hat{\mathcal{D}}^c_j|}{|\hat{\mathcal{D}}^c|}$ inside the absolute value where $|\hat{\mathcal{D}}^c_j|$ terms cancel out:

$$\sum_{j=1}^{J} \left| \frac{\sum_{k \in \hat{\mathcal{D}}^c_j} \hat{p}_k}{|\hat{\mathcal{D}}^c|} - \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{|\hat{\mathcal{D}}^c|} \right|. \quad \text{(A.38)}$$

Splitting $\sum_{\hat{b}_k \in \hat{\mathcal{D}}_j} \hat{p}_k$ for true positives and false positives as $\sum_{\hat{b}_k \in \hat{\mathcal{D}}_j, \psi(k)>0} \hat{p}_k$ and $\sum_{\hat{b}_k \in \hat{\mathcal{D}}_j, \psi(k)\leq 0} \hat{p}_k$ respectively, we have

$$\sum_{j=1}^{J} \left| \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \hat{p}_k + \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)\leq 0} \hat{p}_k}{|\hat{\mathcal{D}}^c|} - \frac{\sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0,} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)})}{|\hat{\mathcal{D}}^c|} \right|. \quad \text{(A.39)}$$

Considering that Eq. (A.39) is minimized when the error for each bin $j$ is minimized as 0, we now focus on a single bin $j$. Note also that for each bin $j$, $|\hat{\mathcal{D}}^c|$ is a constant. As a result, minimizing the following expression minimizes the error for each bin, and also LaECE,

$$\left| \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \hat{p}_k + \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)\leq 0} \hat{p}_k - \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0,} \mathrm{IoU}(\hat{b}_k, b_{\psi(k)}) \right|. \quad \text{(A.40)}$$

By rearranging the terms, we have

$$\left| \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)>0} \Big( \hat{p}_k - \mathrm{IoU}(\hat{b}_k, b_{\psi(k)}) \Big) + \sum_{\hat{b}_k \in \hat{\mathcal{D}}^c_j, \psi(k)\leq 0} \hat{p}_k \right|, \quad \text{(A.41)}$$

which reduces to Eq. (5) once by setting $\hat{p}_k$ by $t^{cal}_k$. This concludes the derivation and validates how we construct the targets $t^{cal}_k$ while obtaining the pairs to train the calibrator.

Table A.13. Dummy detections decrease LaECE superficially with no effect on AP due to top-$k$ survival. LRP Error penalizes dummy detections and requires the detections to be thresholded properly. COCO val set is used.

| Detector | Dummy det. | det/img. | LaECE ↓ | AP ↑ | LRP ↓ |
|---|---|---|---|---|---|
| F-RCNN | None | 33.9 | 15.1 | 39.9 | 86.5 |
| | up to 100 | 100 | 3.9 | 39.9 | 96.8 |
| | up to 300 | 300 | 1.4 | 39.9 | 98.8 |
| | up to 500 | 500 | 0.9 | 39.9 | 99.2 |
| ATSS | None | 86.4 | 7.7 | 42.8 | 95.1 |
| | up to 100 | 100 | 6.0 | 42.8 | 96.2 |
| | up to 300 | 300 | 1.8 | 42.8 | 98.9 |
| | up to 500 | 500 | 1.1 | 42.8 | 99.3 |

### D.4. More Examples of Reliability Diagrams

We provide more examples of reliability diagrams in Fig. A.14 on F-RCNN and ATSS on SAOD-AV. To provide insight on how the error on the set that the calibrator is trained with, Fig. A.14 (a-c) show the reliability diagrams on the val set as the split used to train the calibrator. On the val set, we observe that the isotonic regression method for calibration results in an LaECE of 0.0; thereby overfitting to the training data (Fig. A.14(b)).

On the other hand, the linear regression method ends up with a training LaECE of 5.0 (Fig. A.14(c)). Consequently, we observe that linear regression performs slightly better than isotonic regression on the BDD45K test split (Fig. A.14 (e,f)). Besides, when we compare Fig. A.14(e,f) with Fig. A.14(d), we observe that both isotonic regression and linear regression decrease the overconfidence of the baseline F-RCNN Fig. A.14. As a different type of calibration error, ATSS shown in Fig. A.14(g) is under-confident. Again, linear regression and isotonic regression improves the calibration performance of ATSS. This further validates on SAOD-AV that such post-hoc calibration methods are effective.

### D.5. Numerical Values of Fig. 5

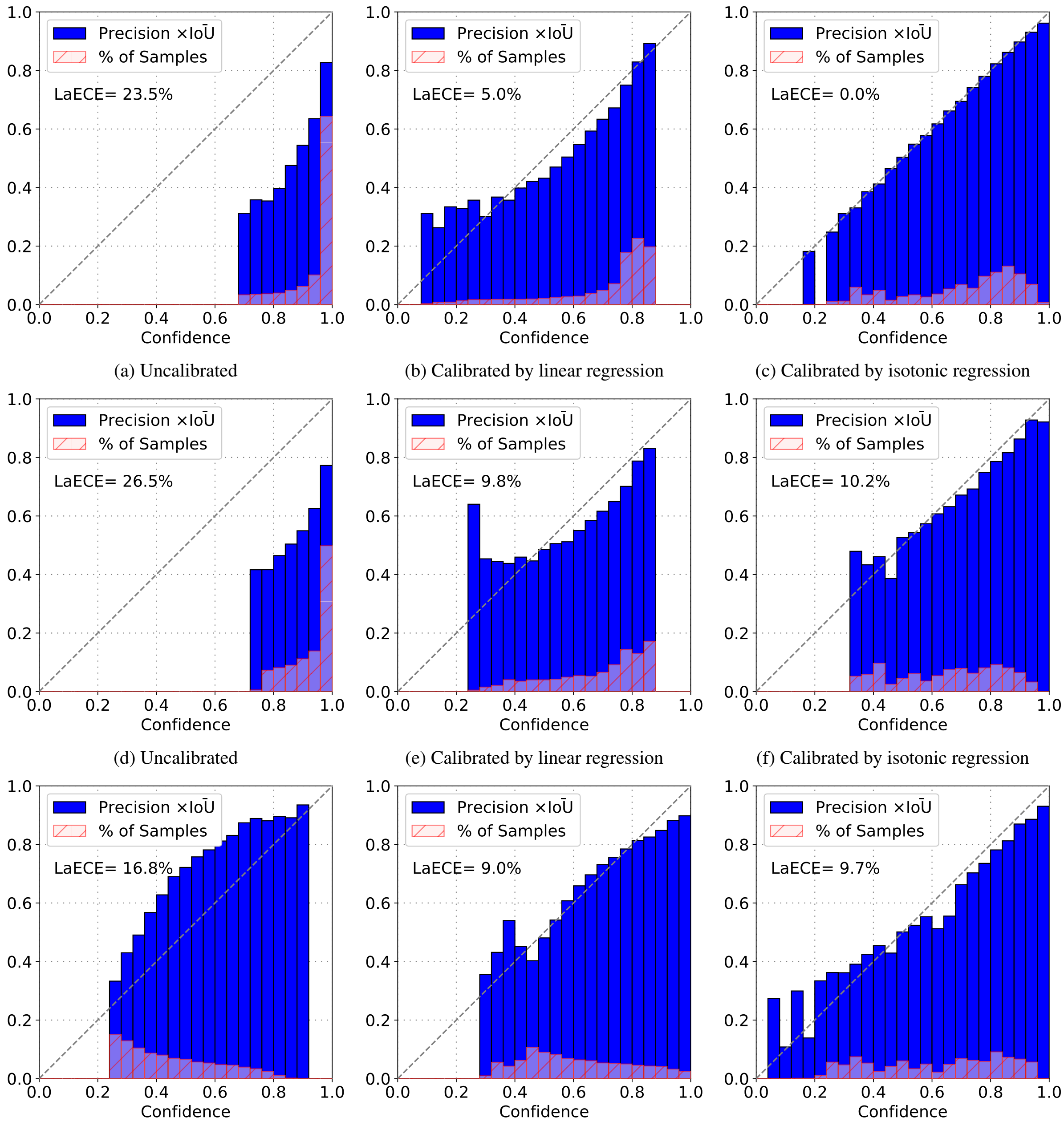
Tables A.13 and A.14 present the numerical values used in the Fig. 5(a) and Fig. 5(b) respectively. Please refer to Sec. 5.2 for the details of the tables and discussion.

## E. Further Details on SAOD and SAODETS

This section provides further details and analyses on the SAOD task and the SAODETS.

### E.1. Algorithms to Make an Object Detector Self-Aware

In Sec. 6, we summarized how we convert an object detector into a self-aware one. Specifically, to do so, we use mean(top-3) and obtain an uncertainty threshold $\bar{u}$ through



(a) Uncalibrated (b) Calibrated by linear regression (c) Calibrated by isotonic regression

(d) Uncalibrated (e) Calibrated by linear regression (f) Calibrated by isotonic regression

(g) Uncalibrated (h) Calibrated by linear regression (i) Calibrated by isotonic regression

Figure A.14. (First row) Reliability diagrams of F R-CNN on SAOD-AV $\mathcal{D}_{\text{Val}}$, which is used to obtain the set that we used for training the calibrators. (Second row) Reliability diagrams of F R-CNN on SAOD-AV $\mathcal{D}_{\text{ID}}$ (BDD45K). (Third row) Reliability diagrams of ATSS on SAOD-AV $\mathcal{D}_{\text{ID}}$ (BDD45K). Linear regression and isotonic regression improve the calibration performance of both over-confidence F-RCNN (compare (e) and (f) with (d)) and under-confidence ATSS (compare (h) and (i) with (g)).

cross-validation using pseudo OOD set approach (Sec. 4), obtain the detections through LRP-optimal thresholding (Sec. 5.2) and calibrate the detection scores using linear regression as discussed in Sec. 5.3. Here, we present the following two algorithms to include the further details on how we incorporate these features into a conventional object detector:

1. *The algorithm to make an object detector self-aware in Alg. A.1*. The aim of Alg. A.1 is to obtain
   - the image-level uncertainty threshold $\bar{u}$;
   - the detection confidence score thresholds for

Table A.14. To avoid superficial LaECE gain (Table A.13); we adopt LRP Error that requires the detections to be thresholded properly. We use LRP-optimal thresholding to obtain class-wise thresholds. Results are from COCO val set. with 7.3 objects/image on average.

| Detector | Threshold | det/img. | LaECE ↓ | AP ↑ | LRP ↓ |
|---|---|---|---|---|---|
| F-RCNN | None | 33.9 | 15.1 | 39.9 | 86.5 |
| | 0.30 | 11.2 | 27.5 | 38.0 | 67.6 |
| | 0.50 | 7.4 | 27.6 | 36.1 | 62.1 |
| | 0.70 | 5.2 | 24.5 | 33.2 | 61.5 |
| | LRP-opt. | 6.1 | 26.1 | 34.6 | 61.1 |
| ATSS | None | 86.4 | 7.7 | 42.8 | 95.1 |
| | 0.30 | 5.2 | 20.2 | 35.3 | 60.5 |
| | 0.50 | 2.0 | 26.6 | 19.7 | 78.4 |
| | 0.70 | 0.3 | 12.3 | 3.9 | 96.3 |
| | LRP-opt. | 6.0 | 18.3 | 36.7 | 60.2 |

**Algorithm A.1** Making an object detector self-aware

```
1:  procedure MakingSelfAware($\mathcal{D}_{\text{Train}}$, $\mathcal{D}_{\text{Val}}$)
2:      Train a standard detector $f(\cdot)$ on $\mathcal{D}_{\text{Train}}$
3:      Obtain pseudo OOD set $\mathcal{D}^{-}_{\text{Val}}$ by replacing objects in $\mathcal{D}_{\text{Val}}$ with zeros (Sec. 4)
4:      Remove the images with no objects from $\mathcal{D}_{\text{Val}}$, denote the resulting set by $\mathcal{D}^{+}_{\text{Val}}$
5:      Make inference on $\mathcal{D}^{-}_{\text{Val}}$ by including top-100 detections from each image, i.e., $\mathcal{D}^{-}_{val} = \{f(X_i)\}_{X_i \in \hat{D}^{-}_{100}}$
6:      Make inference on $\mathcal{D}^{+}_{\text{Val}}$ by including top-100 detections from each image, i.e., $\mathcal{D}^{+}_{val} = \{f(X_i)\}_{X_i \in \hat{D}^{+}_{100}}$
7:      Cross-validate $\bar{u}$, the image-level uncertainty threshold, on $\mathcal{D}^{+}_{val}$ and $\mathcal{D}^{-}_{val}$ using mean(top-3) of the uncertainty scores against the Balanced Accuracy as the performance measure (Sec. 4)
8:      Cross-validate $\bar{v}^c$, the detection-level threshold of class $c$, on $\hat{D}^{+}_{100}$ using LRP-optimal thresholding (Sec. 5.2)
9:      Remove all detections of class $c$ in $\hat{D}^{+}_{100}$ with score less than $\bar{v}^c$ to obtain thresholded detections $\hat{D}_{thr}$
10:     Using $\hat{D}_{thr}$, train a linear regression calibrator $\zeta^c(\cdot)$ for each class $c$ (Sec. 5.3)
11:     return $f(\cdot)$, $\bar{u}$, $\{\bar{v}^c\}_{c=1}^{C}$, $\{\zeta^c(\cdot)\}_{c=1}^{C}$
12: end procedure
```

each class $\{\bar{v}^c\}_{c=1}^{C}$;

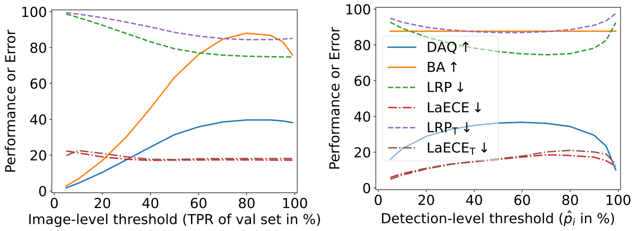
- the calibrators for each class $\{\zeta^c(\cdot)\}_{c=1}^{C}$; and
- a conventional object detector $f(\cdot)$ on which these features will be incorporated into.

To do so, after training the conventional object detector $f(\cdot)$ using $\mathcal{D}_{\text{Train}}$ (line 2), we first obtain the image-level uncertainty threshold $\bar{u}$ by using our pseudo OOD set approach as described in Sec. 4 (lines 3-7). While doing that, we do not apply detection-level thresholding yet; ensuring us to have at least 3 detections from each image on which we compute the image-level uncertainty using mean(top-3) of the uncertainty scores. While we enforce this by keeping a maximum of 100 detections following AP-based evaluation, we only use the top-3 scoring detections to compute the image-level uncertainty. Then, after cross-validating $\bar{u}$, we cross-validate $\{\bar{v}^c\}_{c=1}^{C}$ for each class (line 8) and using only the thresholded detections we train the calibrators (lines 9-10). This procedure allows us to incorporate necessary features into a conventional object detector, making it self-aware.

2. *The inference algorithm of a* SAODet *in Alg.* A.2. Given a SAODet, the inference on a given image $X$ is as follows. We first compute the uncertainty of the detector on image $X$, denoted by $\mathcal{G}(X)$, following the same method in Alg. A.1, that is mean(top-3) (lines 2-3). Then, the rejection or acceptance decision is made by comparing $\mathcal{G}(X)$ by the cross-validated threshold $\bar{u}$ (line 5). If the image is rejected then, no detection is returned (line 7). Otherwise, if the detection is accepted, then its confidence is compared against the cross-validated detection confidence threshold (line 11); enabling us to differentiate between a useful and a low-scoring noisy detection. If the confidence of the detection is larger than the detection-level threshold, then it is added into the final set of detections also by a calibrated confidence score (line 12). Therefore, Alg. A.2 checks whether the detector is able to make a detection on the given image $X$, if so, it preserves the accurate detections obtained by the object detector by removing the noisy detections as well as calibrates the detection scores; applying all features of a SAODet during inference.

**Algorithm A.2** The inference algorithm of a SAODet given an image $X$ (please also refer to Alg. A.1 for the notation)

```
1:  procedure Inference($f(\cdot)$, $\bar{u}$, $\{\bar{v}^c\}_{c=1}^{C}$, $\{\zeta^c(\cdot)\}_{c=1}^{C}$, $X$)
2:      $\hat{D}_{100} = \{\hat{c}_i, \hat{b}_i, \hat{p}_i\}^{N} = f(X)$ such that $N \leq 100$
3:      Estimate $\mathcal{G}(X)$, the image-level uncertainty of $f(X)$ on $X$, using mean(top-3) of the uncertainty scores (Sec. 4)
4:      Initialize the thresholded detection set $\hat{D}_{thr} = \emptyset$
5:      if $\mathcal{G}(X) > \bar{u}$ then
6:          $\hat{a} = 0$
7:          return $\{\hat{a}, \hat{D}_{thr}\}$ // REJECT $X$ with $\hat{D}_{thr}$ being $\emptyset$
8:      else
9:          $\hat{a} = 1$
10:         for each detection $\{\hat{c}_i, \hat{b}_i, \hat{p}_i\} \in \hat{D}_{100}$ do
11:             if $\hat{p}_i \geq \bar{v}^{\hat{c}_i}$ then
12:                 $\hat{D}_{thr} = \hat{D}_{thr} \cup \{\hat{c}_i, \hat{b}_i, \zeta^{\hat{c}_i}(\hat{p}_i)\}$
13:             end if
14:         end for
15:         return $\{\hat{a}, \hat{D}_{thr}\}$ // ACCEPT $X$
16:     end if
17: end procedure
```



(a) Image-level uncertainty thr. (b) Detection confidence thr.

Figure A.15. The effect of image- and detection-level thresholds. DAQ (blue curve) decreases significantly for extreme cases such as when all images are rejected or all detections are accepted; implying its robustness to such cases. Here, for the sake of analysis simplicity, we use a single confidence score threshold ($\bar{v}$) obtained on the final detection scores $\hat{p}_i$ in (b) instead of class-wise approach that we used while building SAODETs.

Table A.15. Effect of common improvements (epochs (Ep.), Multi-scale (MS) training, stronger backbones) on F-RCNN (SAOD-Gen).

| Ep. | MS | Backbone | DAQ | BA | mECE | LRP | mECE$_\text{T}$ | LRP$_\text{T}$ | AP |
|---|---|---|---|---|---|---|---|---|---|
| 12 | ✗ | R50 | 38.5 | 88.0 | **16.4** | 76.6 | **16.8** | 85.0 | 24.8 |
| 36 | ✗ | R50 | 38.4 | 87.4 | 18.7 | 75.9 | 20.5 | 85.0 | 25.5 |
| 36 | ✓ | R50 | 39.7 | 87.7 | 17.3 | 74.9 | 18.1 | 84.4 | 27.0 |
| 36 | ✓ | R101 | 42.0 | **88.1** | 17.5 | 73.4 | 19.0 | 82.8 | 28.7 |
| 36 | ✓ | R101-DCN | **45.9** | 87.4 | 17.3 | **70.8** | 19.4 | **79.7** | **31.8** |

## E.2. Sensitivity of the SAOD Performance Measures to the Image-level Uncertainty Threshold and Detection Confidence Threshold

Here, we explore the sensitivity of the performance measures used in our SAOD framework to the image-level uncertainty threshold $\hat{u}$ and the detection confidence threshold $\bar{v}$. To do so, we measure DAQ, BA, LRP and LaECE of F-RCNN on $\mathcal{D}_{\text{Test}}$ of SAOD-Gen by systematically varying (i) image-level uncertainty threshold $\bar{u} \in [0, 1]$ and (ii) detection-level confidence score threshold $\bar{v} \in [0, 1]$. Note that in this analysis, we do not use LRP-optimal threshold for detection-level thresholding, which obtains $\bar{v}$ for each class but instead employ a single threshold for all classes; enabling us to change this threshold easily. Fig. A.15 shows how there performance measures change for different image-level and detection-level thresholds. First, we observe that it is crucial to set both thresholds properly to achieve a high DAQ. More specifically, rejecting all images or accepting all detections $\bar{v} = 0$ in Fig. A.15 results in a very low DAQ, highlighting the robustness of DAQ in these extreme cases. Second, setting a proper uncertainty threshold is also important for a high BA (Fig. A.15(a)), while it is not affected by detection-level threshold (Fig. A.15(b)) since BA indicates the OOD detection performance but not related to the accuracy or calibration performance of the detections. Third, accepting more images results in better LRP values (green and purple curves in Fig. A.15(a)) as otherwise the ID images are rejected with the detection sets being empty. Similarly, to achieve a high LRP, setting the detection confidence threshold properly is important as well. This is because, a small confidence score threshold implies more FPs, conversely a large threshold can induce more FNs. Finally, while we don't observe a significant effect of the uncertainty threshold on LaECE, a large number of detections due to a smaller detection confidence threshold has generally a lower LaECE.. This is also related to our previous analysis in Sec. 5.2, which we show that more detections imply a lower LaECE as depicted in Fig. A.15(b) when the threshold approaches 0. However, in that case, LRP Error goes to 1, and as a result, DAQ significantly decreases; thereby preventing setting the threshold to a lower value to superficially boost the overall performance.

## E.3. Effect of common improvement strategies on DAQ

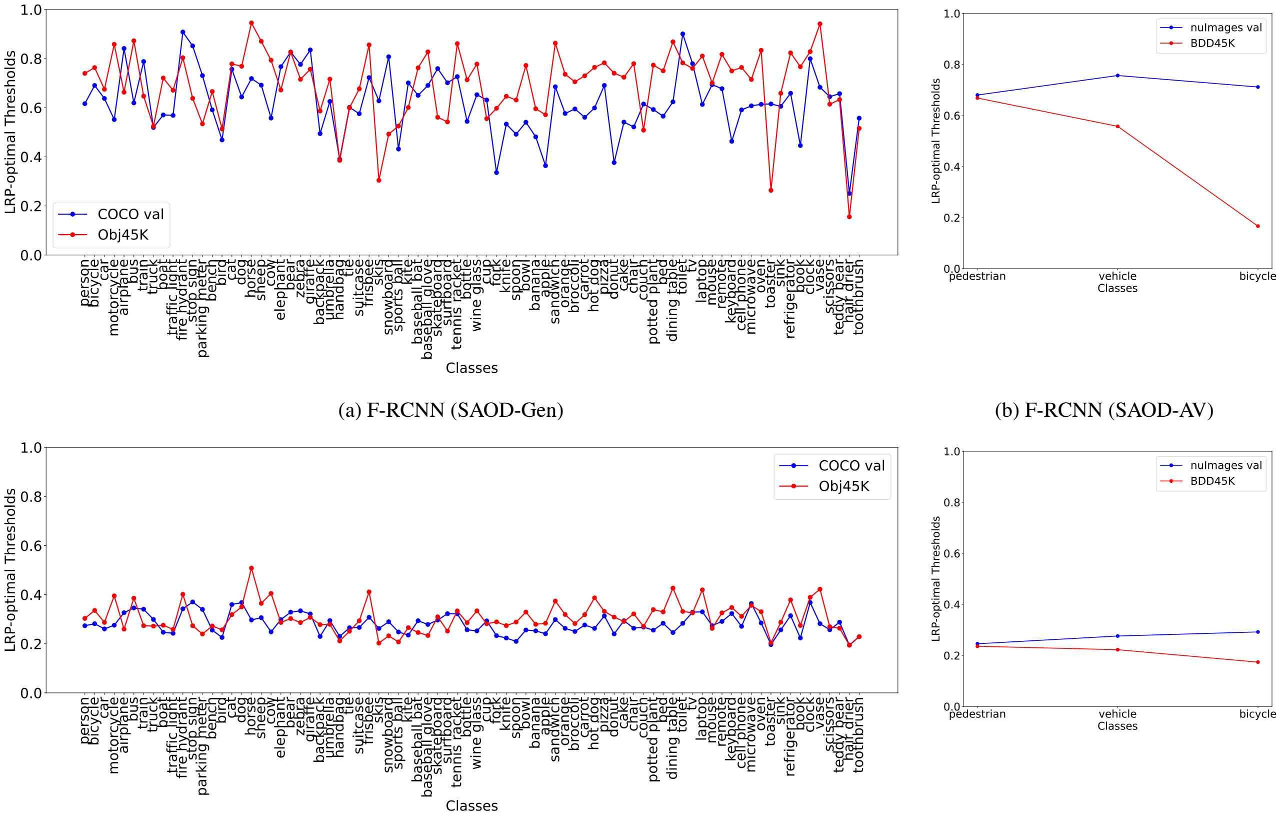
Here, we analyse how common improvement strategies of object detectors affect the DAQ in comparison with AP. To do so, we first use a simple but a common baseline model: We use F-RCNN (SAOD-Gen) trained for 12 epochs without multi-scale training. Then, gradually, we include the following four common improvement strategies commonly used for object detection [6, 55, 77]:

1. increasing number of training epochs,
2. using multiscale training as described in Sec. B,
3. using ResNet-101 as a stronger backbone [22], and
4. using deformable convolutions [78].

Table A.15 shows the effect of these improvement strategies, where we see that stronger backbones increase DAQ, but mainly due to an improvement in LRP Error. It is also worth highlighting that more training epochs improves AP (e.g. going from 12 to 36 improves AP from 24.8 to 25.5), but not DAQ due to a degradation in LAECE. This is somewhat expected, as longer training improves accuracy, but drastically make the models over-confident [47].

## E.4. The Impact of Domain-shift on Detection-level Confidence Score Thresholding

For detection-level confidence score thresholding, we employ LRP-optimal thresholds by cross-validating a threshold $\bar{v}$ for each class using $\mathcal{D}_{\text{Val}}$ against the LRP Error. While LRP-optimal thresholds are shown to be useful if the test set follows the same distibution of $\mathcal{D}_{\text{Val}}$, we note that our $\mathcal{D}_{\text{ID}}$ is collected from a different dataset, introducing domain shift as discussed in App. A. As a result, here



(a) F-RCNN (SAOD-Gen)

(b) F-RCNN (SAOD-AV)

(c) ATSS (SAOD-Gen)

(d) ATSS (SAOD-AV)

Figure A.16. Comparison of (i) LRP-optimal thresholds obtained on $\mathcal{D}_{\text{Val}}$ as presented and used in the paper (blue lines); and (ii) LRP-optimal thresholds obtained on $\mathcal{D}_{\text{ID}}$ as oracle thresholds (red lines). Owing to the domain shift between $\mathcal{D}_{\text{Val}}$ and $\mathcal{D}_{\text{ID}}$, the optimal thresholds do not match exactly. The thresholds between $\mathcal{D}_{\text{Val}}$ and $\mathcal{D}_{\text{ID}}$ are relatively more similar for SAOD-Gen compared to SAOD-AV.

Table A.16. Evaluating self-aware object detectors. In addition to Tab. 6, this table includes the components of the LRP Error for more insight. Particularly, $\text{LRP}_{\text{Loc}}$, $\text{LRP}_{\text{FP}}$, $\text{LRP}_{\text{FN}}$ correspond to the average 1-IoU of TPs, 1-precision and 1-recall respectively.

| | Self-aware Detector | DAQ↑ | $\mathcal{D}_{\text{OOD}}$ vs. $\mathcal{D}_{\text{ID}}$ | | | $\mathcal{D}_{\text{ID}}$ | | | | | | $\mathcal{T}(\mathcal{D}_{\text{ID}})$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | BA↑ | TPR↑ | TNR↑ | IDQ↑ | LaECE↓ | LRP↓ | $\text{LRP}_{\text{Loc}}$↓ | $\text{LRP}_{\text{FP}}$↓ | $\text{LRP}_{\text{FN}}$↓ | IDQ↑ | LaECE↓ | LRP↓ | $\text{LRP}_{\text{Loc}}$↓ | $\text{LRP}_{\text{FP}}$↓ | $\text{LRP}_{\text{FN}}$↓ |
| Gen | SA-F-RCNN | 39.7 | 87.7 | **94.7** | 81.6 | 38.5 | 17.3 | 74.9 | 20.4 | 48.5 | 52.3 | 26.2 | 18.1 | 84.4 | 21.9 | 52.2 | 72.4 |
| | SA-RS-RCNN | 41.2 | **88.9** | 92.8 | 85.3 | 39.7 | 17.1 | 73.9 | 19.3 | 47.8 | 51.9 | 27.5 | **17.8** | 83.5 | 20.4 | 50.8 | 72.1 |
| | SA-ATSS | 41.4 | 87.8 | 93.1 | 83.0 | 39.7 | 16.6 | 74.0 | **18.5** | 47.8 | 52.8 | 27.8 | 18.2 | 83.2 | **20.2** | 53.2 | 71.1 |
| | SA-D-DETR | **43.5** | **88.9** | 90.0 | **87.8** | **41.7** | **16.4** | **72.3** | 18.8 | **45.1** | **50.7** | **29.6** | 17.9 | **81.9** | 20.4 | **49.6** | **69.4** |
| AV | SA-F-RCNN | 43.0 | **91.0** | 94.1 | **88.2** | 41.5 | 9.5 | 73.1 | 26.3 | 13.2 | 58.1 | 28.8 | 7.2 | 83.0 | 26.7 | 12.2 | 74.7 |
| | SA-ATSS | **44.7** | 85.8 | **95.9** | 77.6 | **43.5** | **8.8** | **71.5** | **25.9** | 14.2 | **55.7** | **30.8** | **6.8** | **81.5** | **26.0** | 14.3 | **72.5** |

we investigate whether the detection-level confidence score threshold is affected from domain shift.

To do so, we compute the LRP-optimal thresholds on both $\mathcal{D}_{\text{Val}}$ and $\mathcal{D}_{\text{Test}}$ of F-RCNN and ATSS, and then, we compare them in Fig. A.16 for our both datasets. As $\mathcal{D}_{\text{Test}}$ is not available during training, the thresholds obtained on $\mathcal{D}_{\text{Test}}$ correspond to the oracle detection-level confidence score thresholds. We observe in Fig. A.16 that:

- For both of the settings optimal thresholds computed on val and test sets rarely match.

- While the thresholds obtained on $\mathcal{D}_{\text{Val}}$ (COCO val set) and $\mathcal{D}_{\text{ID}}$ (Obj45K) for SAOD-Gen dataset is relatively similar, they are more different for our SAOD-AV dataset, which can be especially observed for the `bicycle` class.

Therefore, due to the domain shift in our datasets, the optimal threshold diverges from $\mathcal{D}_{\text{Val}}$ to $\mathcal{D}_{\text{ID}}$ especially for AV-OD dataset. This is not very surprising due to the challenging nature of BDD100K dataset including images at night and under various weather conditions, which also ensues a significant accuracy drop for this setting (Tab. A.8).

Furthermore, to see how these changes reflect into the performance measures, we include $\text{LRP}_{\text{Loc}}$, $\text{LRP}_{\text{FP}}$, $\text{LRP}_{\text{FN}}$ components of LRP Error that are defined as the average over the localisation errors (1-IoU) of TPs, 1-precision and 1-recall respectively in Tab. A.16. We would normally expect that the precision and the recall errors to be balanced once the detections are filtered out using the LRP-optimal threshold [58]. This is, in fact, what we observe in $\text{LRP}_{\text{FP}}$ and $\text{LRP}_{\text{FN}}$ for the $\mathcal{D}_{\text{ID}}$ of SAOD-Gen setting,. For example for F-RCNN, $\text{LRP}_{\text{FP}} = 48.5$ and $\text{LRP}_{\text{FN}} = 52.3$; indicating a relatively balanced precision and recall errors. As for SAOD-AV setting, the significant domain shift of BDD45K is also reflected in the difference between $\text{LRP}_{\text{FP}}$ and $\text{LRP}_{\text{FN}}$ for both F-RCNN and ATSS. For example for F-RCNN, $\text{LRP}_{\text{FP}} = 13.2$ and $\text{LRP}_{\text{FN}} = 58.1$; indicating a significant gap. Besides, as the domain shift increases with $\mathcal{T}(\mathcal{D}_{\text{ID}})$ on SAOD-AV, the gap between $\text{LRP}_{\text{FP}}$ and $\text{LRP}_{\text{FN}}$ increases more. These suggest that more accurate detection-level thresholding methods are required under domain-shifted data.

### E.5. Qualitative Results of SAODETs in comparison to Conventional Object Detectors

In order provide more insight on the behaviour of SAODETs, here we present the inference of SAODETs in comparison to conventional object detectors. Here, we use SA-ATSS and SA-F-RCNN on SAOD-AV dataset. To be consistent with the evaluation, we plot all the detection boxes as they are evaluated: That is, we use top-$k$ survival for F-RCNN and ATSS and thresholded detections for SA-F-RCNN and SA-ATSS. In the following, we discuss the main features of the SAODETs using Fig. A.17, Fig. A.18, Fig. A.19 and Fig. A.20.

**OOD Detection** Fig. A.17 shows on three different input images from different subsets of $\mathcal{D}_{\text{OOD}}$ that a conventional F-RCNN performs detection on OOD images and output detections with high confidence. For example F-RCNN detects a `vehicle` object with 0.84 confidence on the OOD image from Objects365 (last row). On the other hand, SA-F-R-CNN can successfully leverage uncertainty estimates to reject these OOD images.

**Calibration** In Fig. A.14, we presented that ATSS is under-confident and F-RCNN is over-confident. Now, Fig. A.18 shows that the calibration performance of these models are improved accordingly. Specifically, SA-ATSS now has larger confidence scores compared to its conventional version and vice versa for SA-F-RCNN. This can enable the subsequent systems exploiting these confidence scores for decision making to abstract away the difference of the confidence score distributions of the detectors.

**Removing Low-scoring Noisy Detections** We previously discussed in Sec. 5.2 that the detections obtained with top-$k$ survival allows low-scoring noisy detections and that the performance measure AP promotes them (mathematical proof in App. D). This is also presented in the images of conventional object detectors in Fig. A.18. For example, the output of ATSS includes several low-scoring detections, which the practical applications might hardly benefit from. On the other hand, the outputs of the SAODETs in the same figure are more similar to the objects presented in ground truth images (first column) and barely contain any detection that may not be useful.

**Domain Shift** Fig. A.19 includes images with the corruptions with severities 1, 3 and 5. Following the design of the SAOD task, SA-F-RCNN accepts the images with corruptions 1 and 3 and provide detections also by calibrating the detection scores which is similar to $\mathcal{D}_{\text{ID}}$. However, for the image with severity 5, different from the conventional detector, SA-F-CNN rejects the image; implying that the detector is uncertain on the scene.

**Failure Cases** Finally in Fig. A.20, we provide images that SA-F-RCNN and SA-ATSS fail to identify the image from $\mathcal{D}_{\text{OOD}}$ as OOD, but instead perform inference.

### E.6. Suggestions for Future Work

Our framework provides insights into the various elements needed to build self-aware object detectors. Future research should pay more attention to each elements independently while keeping in mind that these elements are tightly intertwined and greatly impact the ultimate goal. One could also try to build a self-aware detector by directly optimizing DAQ which accounts for all the elements together, although in its current state it is not differentiable so a proxy loss or a method to differentiate through such non-differentiable functions would need to be employed.

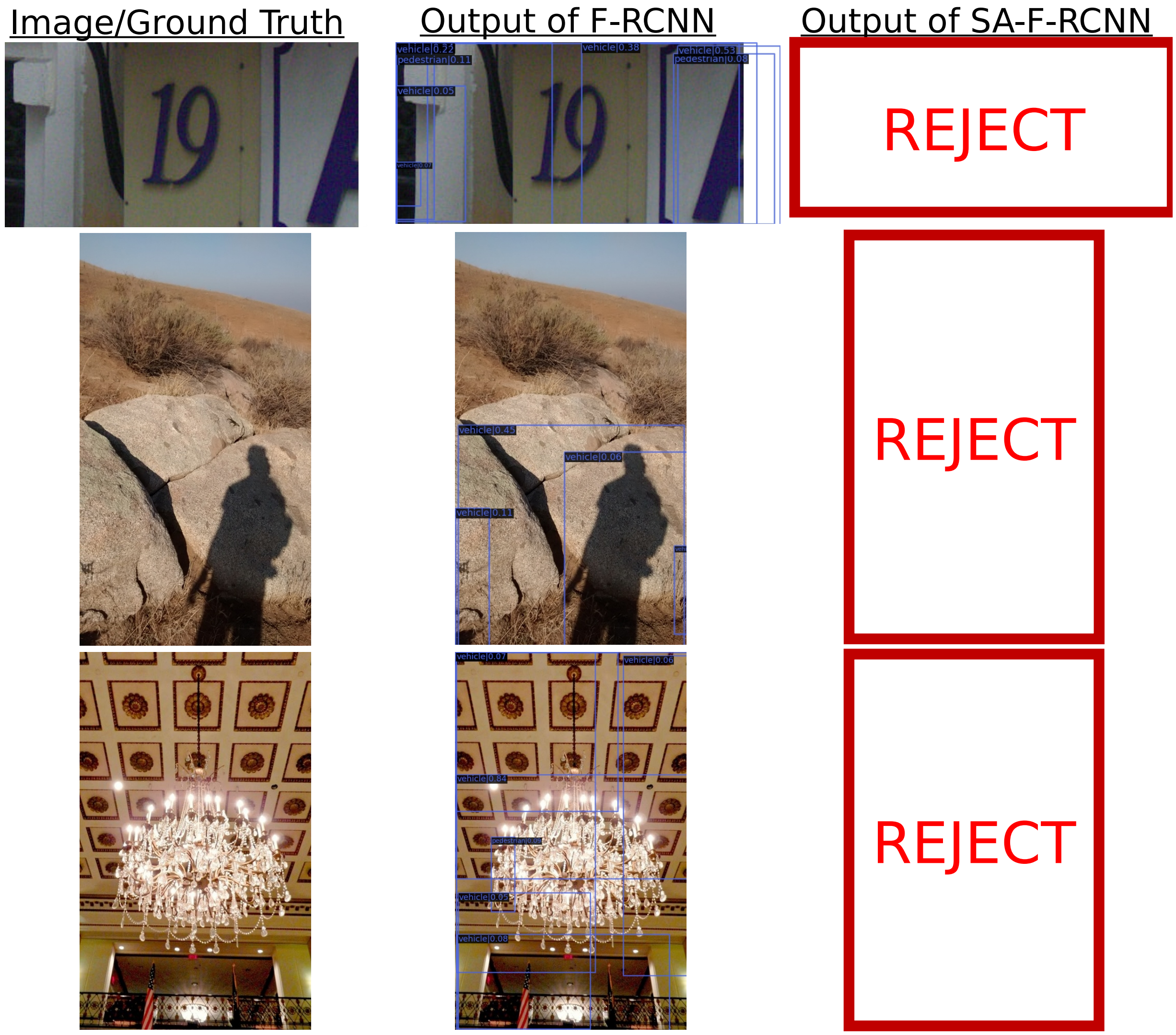


Figure A.17. Qualitative Results of F-RCNN vs. SA-F-RCNN on $\mathcal{D}_{\text{OOD}}$. The images in the first, second and third rows correspond SVHN, iNaturalist and Objects365 subset of $\mathcal{D}_{\text{OOD}}$. While F-RCNN performs inference with non-empty detections sets, SA-F-RCNN rejects all of these images properly.

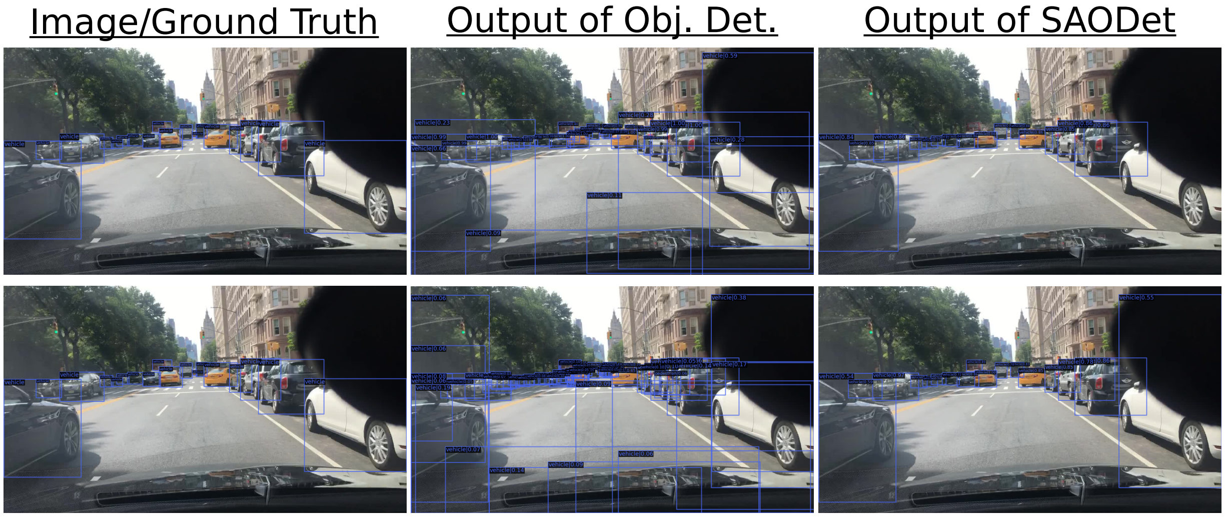


Figure A.18. Qualitative Results of Object detectors and SAODETs on $\mathcal{D}_{\text{ID}}$. (First row) F-RCNN vs. SA-F-RCNN. (Second row) ATSS vs. SA-ATSS. See text for discussion. The class labels and confidence scores of the detection boxes are visible once zoomed in.

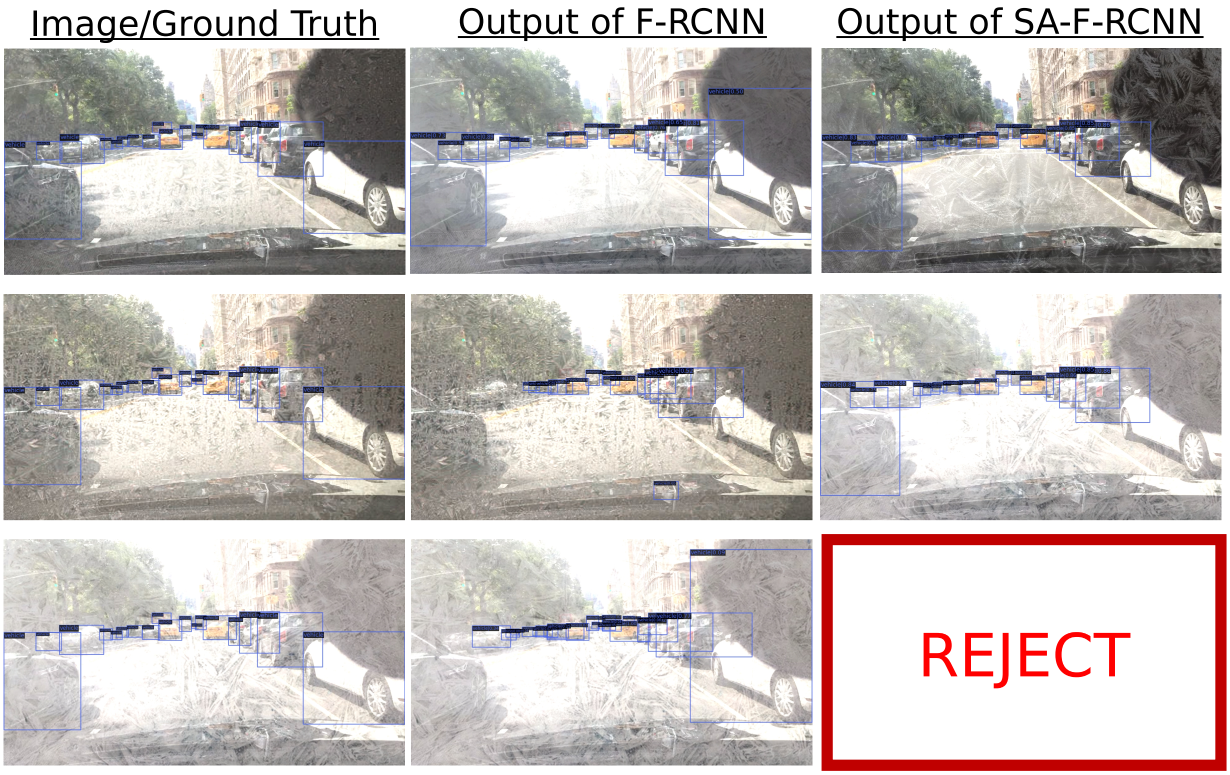


Figure A.19. Qualitative Results of F-RCNN vs. SA-F-RCNN on $\mathcal{T}(\mathcal{D}_{\text{ID}})$ using SAOD-AV dataset. First to third row includes images from $\mathcal{T}(\mathcal{D}_{\text{ID}})$ in severities 1, 3 and 5 as we used in our experiments. The class labels and confidence scores of the detection boxes are visible once zoomed in. For each detector, we sample a transformation using the 'frost' corruption.

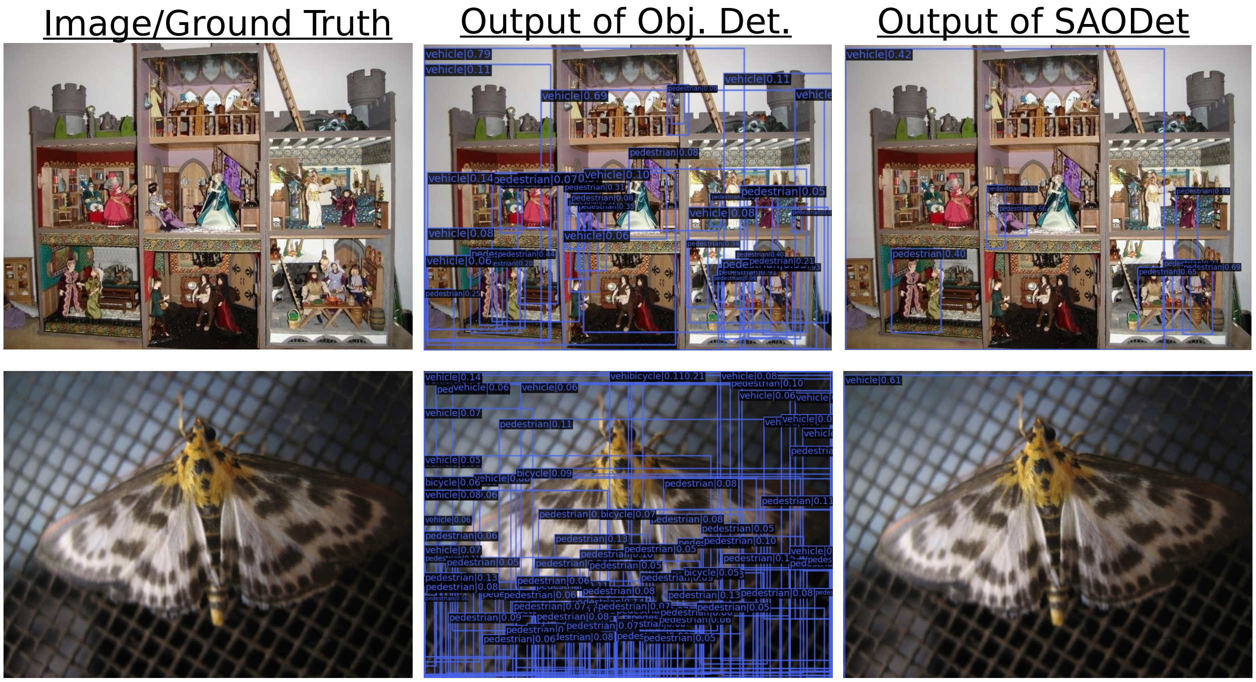


Figure A.20. Failure cases of SAODets in comparison to Object detector outputs. First row includes an image from iNaturalist subset of $\mathcal{D}_{\text{OOD}}$ with the detections from ATSS and SA-ATSS trained on nuImages following our SAOD-AV dataset. While SA-ATSS removes most of the low-scoring detections, it still classifies the image as ID and perform inference. Similarly, the second row includes an image from Objects365 subset of $\mathcal{D}_{\text{OOD}}$ with the detections from F-RCNN and SA-F-RCNN trained on nuImages again following our SAOD-AV dataset. SA-F-RCNN misclassifies the image as ID and performs inference.