A Implementation Details

In this section, we introduce the implementations details of the models and techniques for improving the robustness in the experiments conducted in Sec. 4.

A.1 Image Classification

For the experiments of image classification on ROBIN datasets, we tested three network architectures, namely, MobileNetV3-Large [5], ResNet-50 [3], and Swin-T [7]. We train all the three models with the same hyper-parameter to make a fair comparison. The Batchsize is set to 64 with a step decayed learning rate initialized with 1e-4 and then multiplied by 15,30,45 epochs, we train the network for a total of 100 epochs on the training set. The resolution of the input images are 224 by 224 which is also a defaulted value for training networks [3].

We compared the effectiveness of different data augmentation techniques, namely, style transfer [2], AugMix [4], and adversarial training [10]. For all the experiments using style transfer [2], we use the code from the original authors [1] to create the style augmented images for training. For experiments with AugMix [4], we adopted a PyTorch-based implementation [2]. For adversarial training, we adopted the implementation from the official source. [3]

A.2 Object Detection

We mainly used two frameworks for the task of object detection, namely Faster-RCNN [8] and RetinaNet [6]. Similarly, we keep all the hyper-parameter the same except for the ones we wish to study. The experiments are mainly conducted using the detectron2 codebase [4]. For strong data augmentation techniques that can be used to improve the robustness of vision models, AugMix [4] is relatively harder to implement than the other on object detection because of the image mixing step, so we only evaluated the performance of style transfer and adversarial training. The style transfer uses the same images generate for image classification, and we followed the same procedure to do the adversarial training for object detection.

We train all the object detection models with 18000 iterations with a initial learning rate of 0.02 and a batchsize of 16, the learning rate is then multiplied by 0.1 at 12000 and 16000 iterations. We adopted the multi-scale training technique to improve the baseline performance, each input images will be resized to have a short edge of $[480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800]$, and when testing, the test input image will be resized to have a short edge of 800. For experiments with Swin-T as the backbone network in the detection framework, we adopted the implementations from the authors of the swin-transformer [5].

[1] https://github.com/rgeirhos/Stylized-ImageNet
[2] https://github.com/psh150204/AugMix
[3] https://github.com/locuslab/fast_adversarial
[4] https://github.com/facebookresearch/detectron2
[5] https://github.com/SwinTransformer/Swin-Transformer-Object-Detection

A.3 3D pose estimation

For 3D pose estimation, we evaluated two types of models, Res50-Specific [11] and NeMo [9]. We adopted the implementation from the original authors [6][7]. When training the pose estimation models, we use a batchsize of 108 and a learning rate of 1e-3. For the pose estimation model for each category, we train the model for 800 epochs.

B Detailed statistics

In Tab. 1, we provide the statistics of our dataset. Note that for chair, diningtable, and sofa, it is difficult to find images with the weather nuisance, so the number of images for these categories with weather nuisances is 0.

Table 1: Number of images in each categories with individual nuisances that we defined.

#img	Shape	Pose	Texture	Context	Weather	Total
aeroplane	27	40	66	79	108	320
bus	83	18	82	4	30	217
car	159	24	40	20	83	326
train	34	42	130	70	66	342
boat	30	82	29	30	76	247
bicycle	64	70	28	78	113	353
motorbike	89	108	76	27	97	397
chair	40	40	42	17	0	139
diningtable	22	65	18	59	0	164
sofa	15	28	24	60	0	127
Total	563	517	535	444	573	2632

C Example Images from ROBIN

We show some example images next page. We will release the full dataset.

D Images filtered from the original PASCAL3D+ dataset

This section shows example images that we filtered out from the original PASCAL3D+ dataset [1] in order to make the ROBIN test set really OOD. The images are removed because they are too similar to the images in the ROBIN test set.

In our anonymous repository, we provide all the images that we removed from the original PASCAL3D+ dataset.

[6] https://github.com/shubhtuls/ViewpointsAndKeypoints
[7] https://github.com/Angtian/NeMo

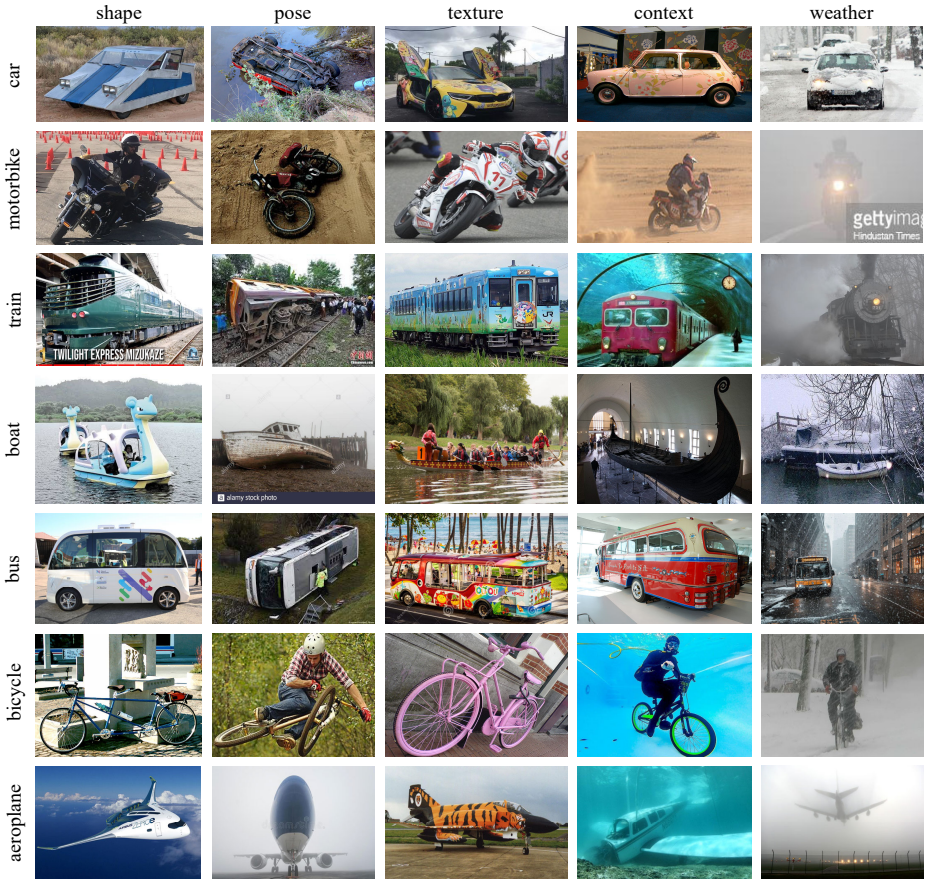

Fig. 1: More example images from ROBIN dataset, we will release the full dataset.

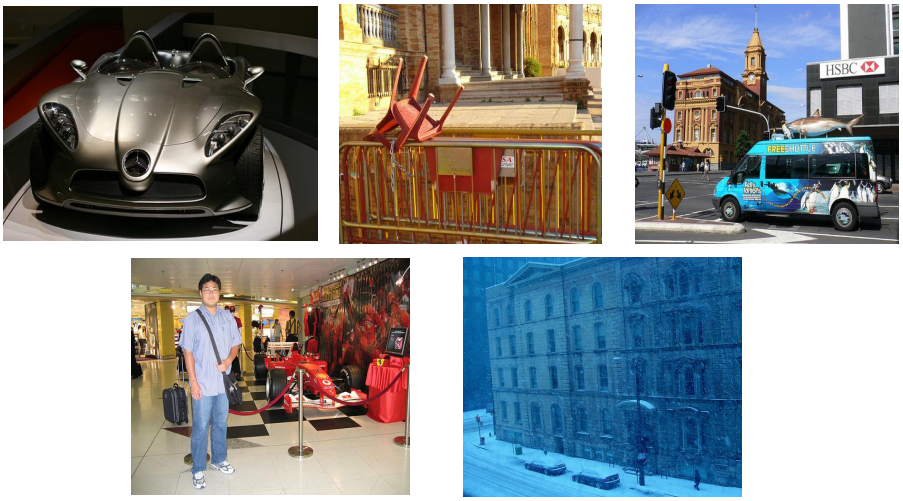

Fig. 2: Example images that are filtered out from the original PASCAL3D+ dataset. These images has nuisances that are similar to the ones we collected in the ROBIN dataset, so they are removed from the training set. We attached all the filtered images with the supplementary.

E Example images with multiple nuisances

We also removed the images that have multiple nuisances from our internet search, we give examples of multiple nuisances in Fig. 3.

F The user interface of our annotation tools

Here we also provide the user interface of our used annotation tools for bounding boxes annotation and 3D pose annotations. The annotation tools are taken and slightly modified from a GitHub project [8] and the original PASCAL3D+ dataset [9]. Identifying informations have been removed from the screenshots.

References

1. Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.*, 2015. 2
2. Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Int. Conf. Learn. Represent.*, 2019. 1
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015. 1

[8] https://github.com/jsbroks/coco-annotator
[9] https://cvgl.stanford.edu/projects/pascal3d.html

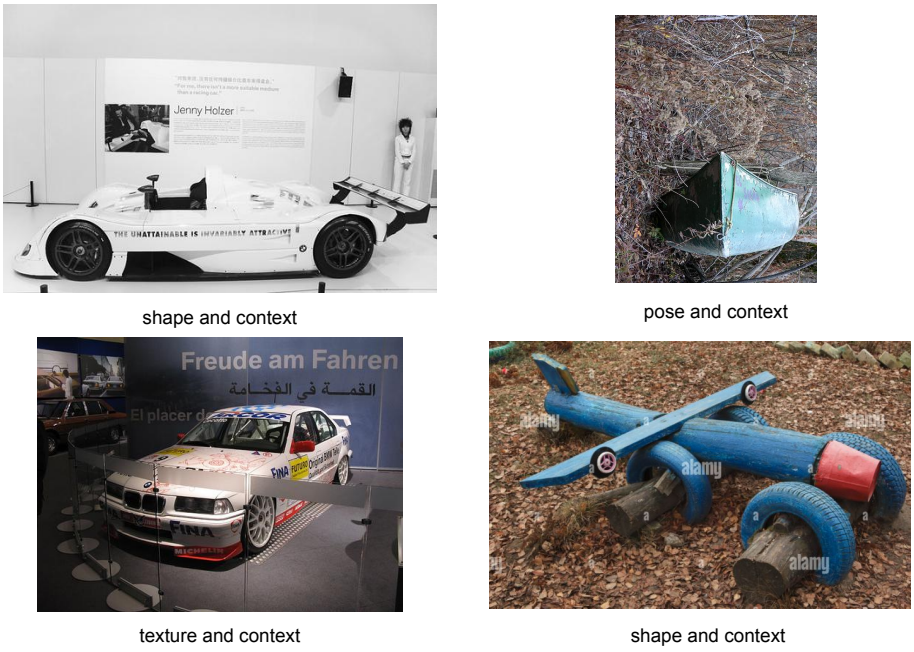

Fig. 3: Example images with multiple nuisance. From our internet search, we also collected many images with multiple nuisance factors, these images are later removed to ensure that we are testing with only one controllable nuisances.

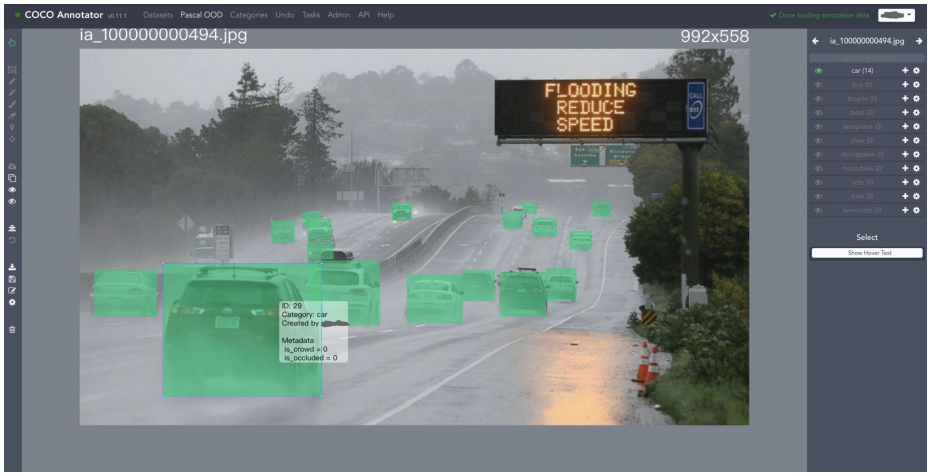

Fig. 4: The user interface of the detection annotation tool.

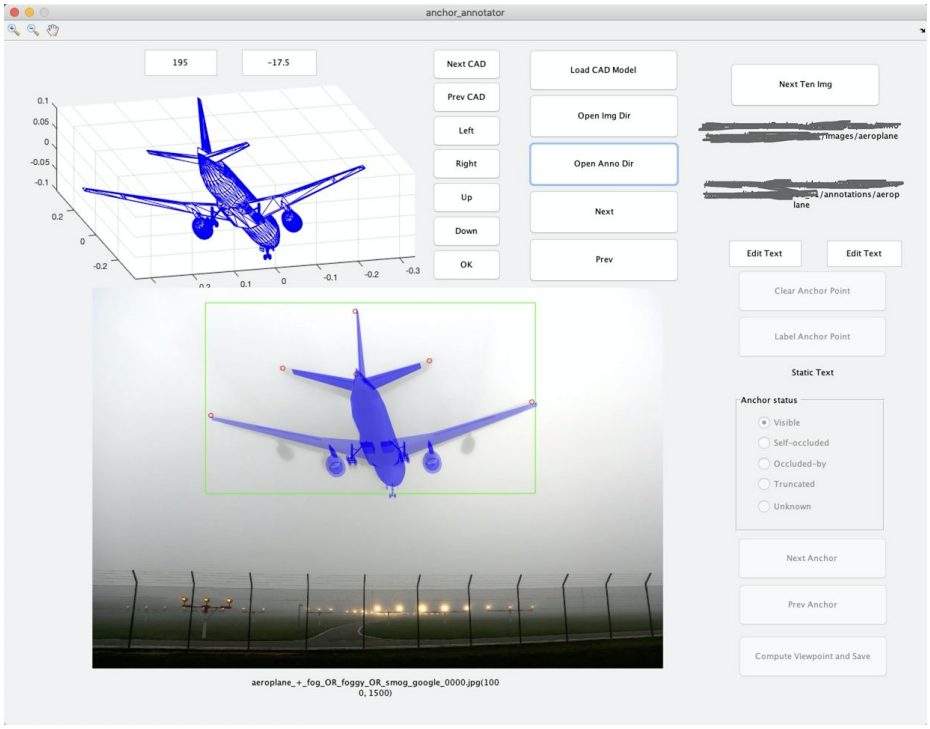

Fig. 5: The user interface of the 3D pose annotation tool.

4. Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *Int. Conf. Learn. Represent.*, 2020. 1
5. Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Int. Conf. Comput. Vis.*, 2019. 1
6. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Int. Conf. Comput. Vis.*, 2017. 1
7. Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, 2021. 1
8. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Adv. Neural Inform. Process. Syst.*, 2015. 1
9. Angtian Wang, Adam Kortylewski, and Alan Yuille. Nemo: Neural mesh models of contrastive features for robust 3d pose estimation. In *Int. Conf. Learn. Represent.*, 2021. 2
10. Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *Int. Conf. Learn. Represent.*, 2020. 1
11. Xingyi Zhou, Arjun Karpur, Linjie Luo, and Qixing Huang. Starmap for category-agnostic keypoint and viewpoint estimation. In *Eur. Conf. Comput. Vis.*, 2018. 2