A. Task Description

Task Description	# Train Images	# Test Images
Object Detection		
Sim10k → Cityscapes	10000	25000
Cityscapes → Foggy	20000	5000
Cityscapes → KITTI	25000	7481
Cityscapes → BDD100k	25000	10000
MS-COCO 4 → 1	94400	23600
MS-COCO 9 → 1	106200	11800
Image Segmentation		
GTA → Cityscapes	25000	2975
Synthia → Cityscapes	9400	2975

Table 1: Description of the tasks, number of train images and number of test images required to train the vanilla unsupervised domain adaptation algorithms, which require access to the test-images at training time [43, 4]

The number of training and testing images for each dataset split is available in Table 1. The number of testing images describes the number of unlabeled images that are available at training time for the baselines "Chen et al." [4] and "Saito et al." [43] for the vanilla unsupervised object detection results, since both methods assume access to the test distribution at training time.

B. Further Experiments

Task	TTT	Tent	Ours
Sim10k → Cityscapes	13.2	15.6	**8.9**
Cityscapes → Foggy Cityscapes	10.4	11.3	**9.0**
Cityscapes → KITTI	10.5	11.8	**8.5**
Cityscapes → BDD100k	9.6	10.2	**8.3**

Table 2: **Detection-Expected Calibration Error (d-ECE)** of the models on the test set. Lower d-ECE is better. All models are trained on the novel distributions with a budget (n) = 64. We see that using TeST, we not only improve the accuracy performance of the models, but also are able to improve the calibration of the models predictions.

Calibration of Predictions One metric that is often overlooked when deploying predictive models is that of calibration: are the probability scores calibrated to their performance. Similar to accuracy, we also analyse the effect of test-time adaptation on the calibration of the resultant models. We use the recent Detection-Expected Calibration Error (D-ECE) metric to measure the calibration of the predictions [26]. To test this, we experiment with the same self-drivimg domain adaptation benchmarks we used for the object detection using a base Faster-RCNN detector. The results are presented in Table 2. Interestingly, we see that using pseudo-labels from the teacher, we are able to outperform both TTT and Tent by being better calibrated. By performing entropy minimization and knowledge distillation on the student, we are able to get better performance, and model predictions that are better calibrated.

C. Qualitative Results

Along with the quantitative results, we perform a thorough qualitative evaluation of TeST. We first further investigation the effect of adding TeST by investigating the true-postives and false-positive outputs from the object detectors. Figures 9 and 10 show results for a base Faster RCNN object detector [38] on the BDD100k [66] and MS-COCO [31] datasets, respectively before and after TeST. Figures 11 and 12 show results for a base Deformable Detection Transformer object detector [70] on the BDD100k [66] and MS-COCO [31] datasets, respectively, before and after TeST. All the images are randomly sampled, and **we do not perform any cherry-picking to get better qualitative results**. We see that in each of the examples, by adding TeST, we are able to increase the number of true positives, while decreasing the number of false positives, both of which are important qualities of a good object detector. By performing qualitative examples on a driving dataset (BDD-100k) and a *common objects* datset (MS-COCO), we are able to evaluate the qualitative improvements on both fronts.

Furthermore, we also investigate the representations learned by TeST by looking at the 3-nearest neighbours in the test set for a given *query image*. Figure 13 and 14 show the 3-nearest neighbour results for the MS-COCO dataset for a Faster RCNN [38] and a Deformable DeTR [70], respectively. The query and neighbour images are from the test-set. We see that using TeST, the model consistently learns semantically relevant features, as the nearest neighbours are from the same class as the objects in the query image.

Figure 9: Qualitative results from BDD100k with a Faster RCNN Object Detector [38]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST have fewer false-positives and more true-positives, which strongly suggests that TeST is able to improve the final object detector on the novel dataset.

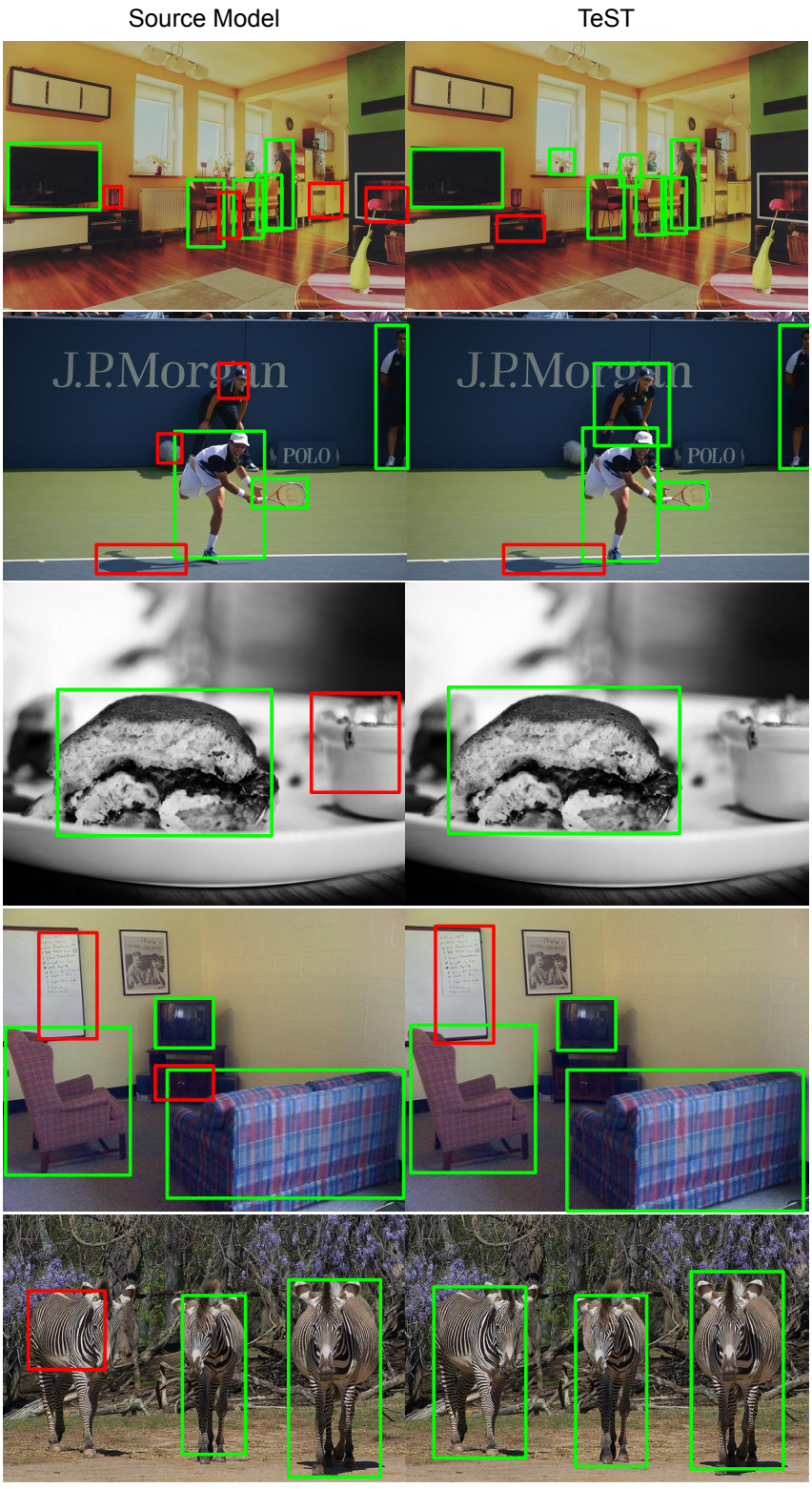

Figure 10: Qualitative results from MS-COCO with a Faster RCNN Object Detector [38]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST have fewer false-positives and more true-positives, which strongly suggests that TeST is able to improve the final object detector on the novel dataset.

Figure 11: Qualitative results from BDD100k with a Deformable DeTR Object Detector [70]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST have fewer false-positives and more true-positives, which strongly suggests that TeST is able to improve the final object detector on the novel dataset. We see that the models trained with TeST have fewer false-positives and more true-positives, which strongly suggests that TeST is able to improve the final object detector on the novel dataset.

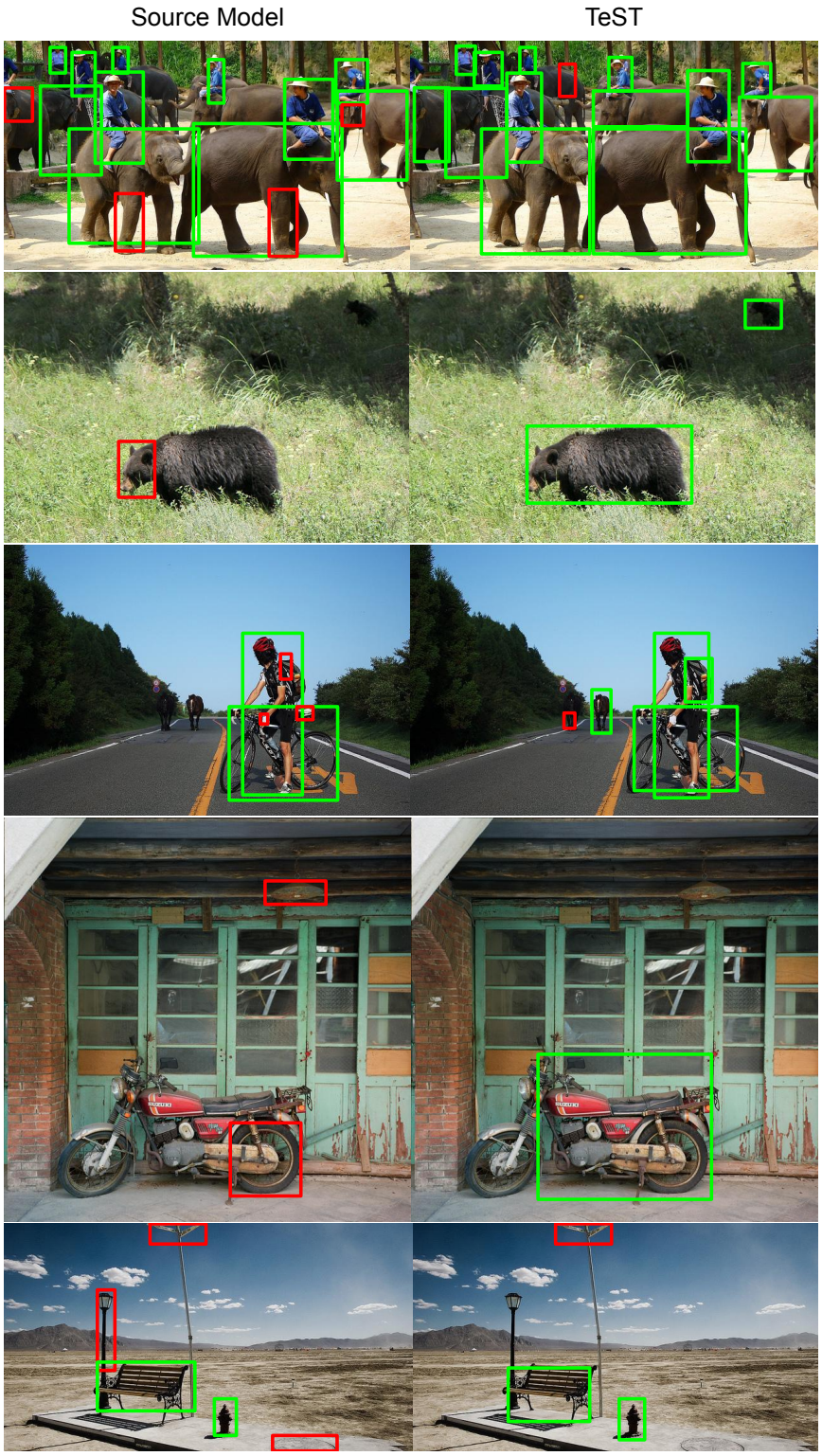

Figure 12: Qualitative results from MS-COCO with a Deformable DeTR Object Detector [70]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST have fewer false-positives and more true-positives, which strongly suggests that TeST is able to improve the final object detector on the novel dataset.

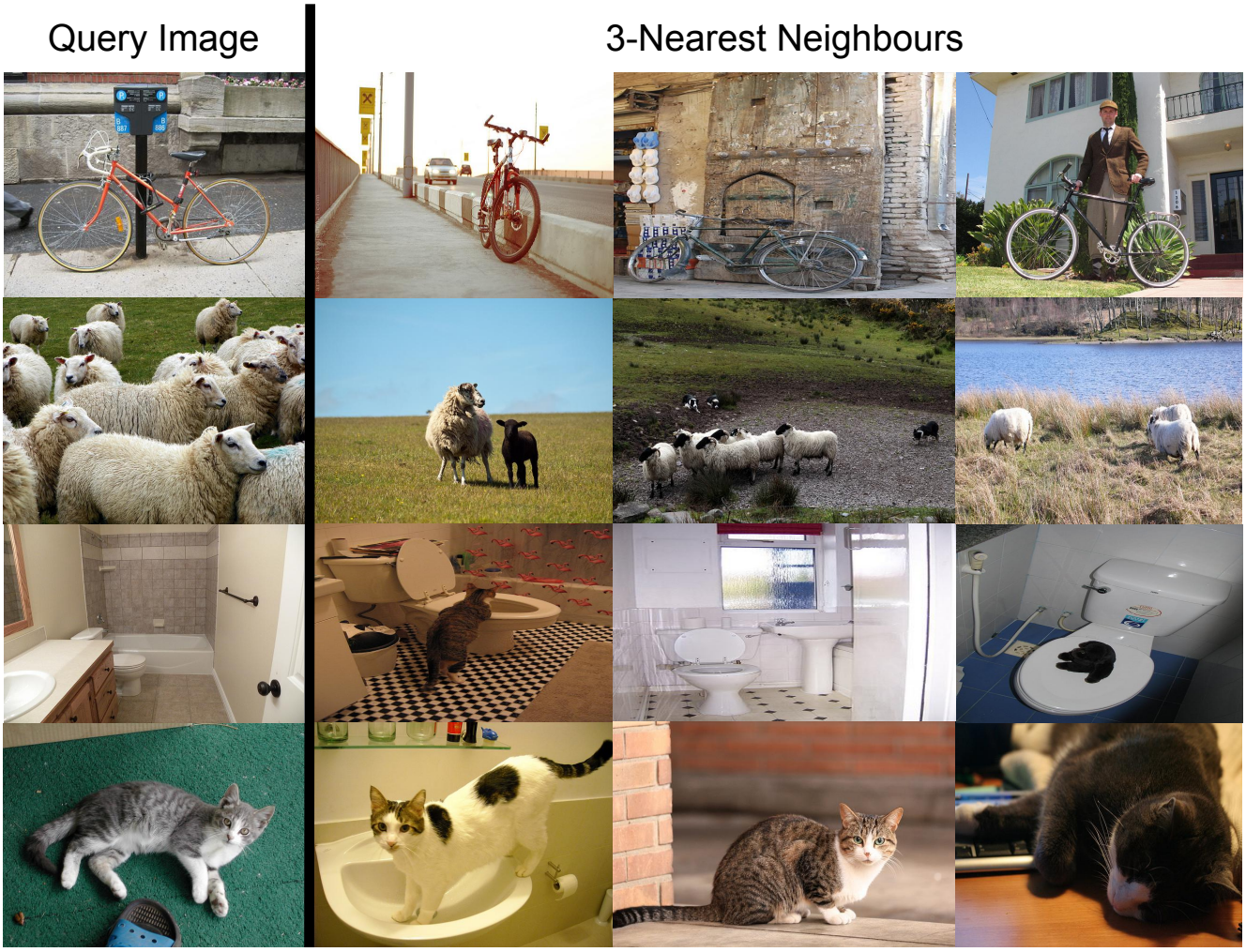

Figure 13: 3-Nearest Neighbours in the embedding space for the feature extractor of a Faster-RCNN detector [38], after it has been trained using TeST on the COCO dataset. We see that the model is able to learn semantically meaningful representations and the nearest neighbours to the query image are semantically similar, thereby showing that TeST is able to perform meaningful reprensentation learning.

Figure 14: 3-Nearest Neighbours in the embedding space for the feature extractor of a Deformable DeTR detector [70], after it has been trained using TeST on the COCO dataset. We see that the model is able to learn semantically meaningful representations and the nearest neighbours to the query image are semantically similar, thereby showing that TeST is able to perform meaningful reprensentation learning.