TeST: Test-time Self-Training under Distribution Shift

Samarth Sinha
University of Toronto

Peter Gehler
AWS Tubingen

Francesco Locatello
AWS Tubingen

Bernt Schiele
AWS Tubingen

Abstract

Despite their recent success, deep neural networks continue to perform poorly when they encounter distribution shifts at test time. Many recently proposed approaches try to counter this by aligning the model to the new distribution prior to inference. With no labels available this requires unsupervised objectives to adapt the model on the observed test data. In this paper, we propose Test-Time Self-Training (TeST): a technique that takes as input a model trained on some source data and a novel data distribution at test time, and learns invariant and robust representations using a student-teacher framework. We find that models adapted using TeST significantly improve over baseline test-time adaptation algorithms. TeST achieves competitive performance to modern domain adaptation algorithms [4, 43], while having access to 5-10x less data at time of adaption. We thoroughly evaluate a variety of baselines on two tasks: object detection and image segmentation and find that models adapted with TeST. We find that TeST sets the new state-of-the art for test-time domain adaptation algorithms.

1. Introduction

Deep learning models have shown excellent promise in computer vision research, where models are used to perform core computer vision tasks such as image classification [48, 18], pixel-wise segmentation [33, 65, 17], and object detection [12, 38, 30]. However, one key limitation still shared with most statistical machine learning models is the limited ability to generalize across distribution shift. In distribution shift we face test data that comes from a different distribution than what the model has been trained on [10, 36]. More specifically, when there is a covariate shift between the training distribution $P_{\mathcal{S}}(x)$, and the test distribution $P_{\mathcal{T}}(x)$, but the distribution of the classes $P(y|x)$ remains constant, neural networks are unable to adapt to such novel domains. This affinity towards the training distribution hinders the ability of the model to be deployed in real-world settings, as it is impossible to train the model on all possible data distributions that it may encounter in the real-world.

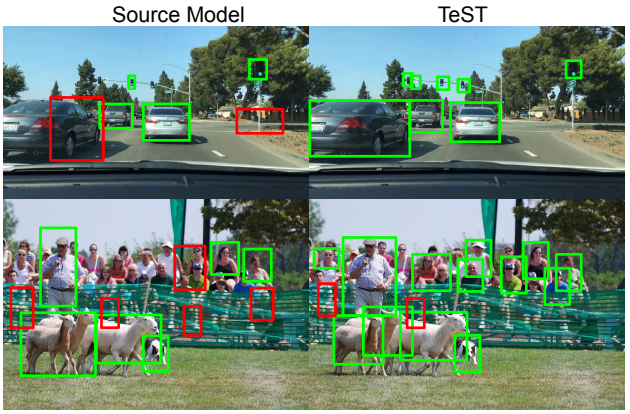


Figure 1: The output of a Faster-RCNN before and after trained on CityScapes (top) and MS-COCO (bottom). Note that the number of false positives (red rectangles) is reduced while the number of true positives (green rectangles) is significantly improved. We see that using TeST, we are able to significantly improve the bounding box positions, and also able to capture more objects that are clearly within the scene. For example, in the top image, the TeST model correctly detects traffic signs and traffic lights far from the car, while also improving the bounding box location for the cars. Similarly, the TeST model is able to detect more people in the background in the bottom example, as well as able to detect more sheep in the scene.

Typically, the problem of *domain adaptation* has been addressed by assuming that the model has access to unlabeled test data during training time [42, 63, 2, 21]. The downside of this strategy is that the model can only adapt to distributions for which data is available during training, and collecting data from all possible target distributions is often impractical. Recently, transductive approaches, also referred to *test-time adaptation*, have garnered significant attention [52, 58, 1]. But such algorithms do not directly solve the problem of learning invariant representations such that the model can deal with distribution shifts as in the case of domain adaptation. Towards this, we propose a test-time

adaptation method that utilizes consistency regularization to learn *invariant representations* [51, 49], self-distillation using a student-teacher framework to learn *robust representations* from the test-data for adaptation [59], and entropy minimization to produce *confident predictions* on the novel test data [58]. We find that despite its simplicity, our Test-time Self Training (TeST) approach significantly improves the performance of the model, as can be seen qualitatively in Figure 1. We achieve this without *any changes to the training paradigm* (such as having to train the source model on an additional image rotation loss at test time [52]) or access to the source data [43, 4], and do not require large batch-size [58]. Our only requirements is that the label space is shared by the source and target domain.

TeST is a two stage procedure to perform test-time adaptation to account for distribution shift. We propose to finetune a trained *source* model using a two-stage self-training procedure which first trains the *teacher* using consistency regularization on the novel data distribution to learn invariant representations. Then we leverage knowledge distillation using the teachers learned representations onto the student; the final student is then evaluated on its ability to generalize to the novel test data. An overview of the method is presented in Figure 2. We thoroughly evaluate our method and all baselines on large-scale self-driving object detection and image segmentation domain adaptation benchmarks, where we are able to significantly outperform all baselines. We observe that TeST consistently outperforms all other test-time adaptation methods on small and large *budgets* of test data available for finetuning. We perform our analysis using 6 challenging domain adaptation tasks for object detection, and 2 challenging tasks for image segmentation. In our results we find that, TeST significantly outperforms other test-time adaptation algorithms, and performs on par with domain adaptation algorithms [4, 43], while requiring 5-10 times fewer target images.

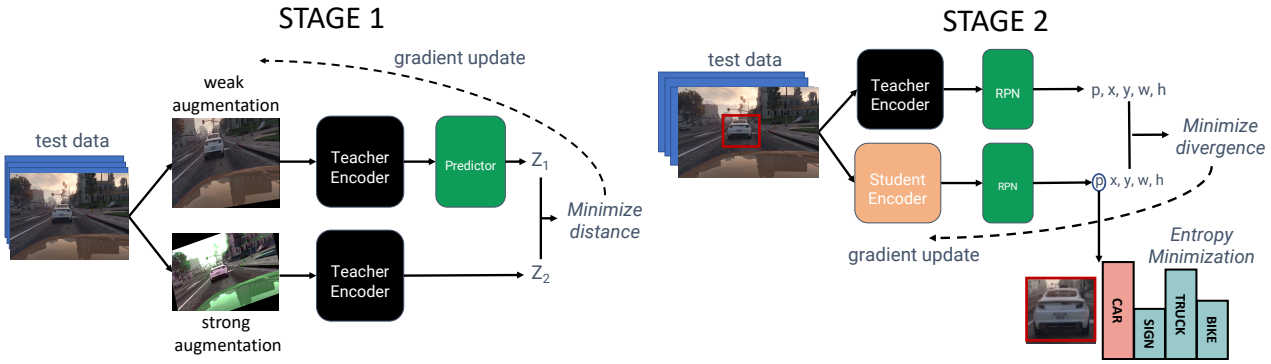
Our main contributions can be summarized as: (1) We propose a simple test-time adaptation system which utilizes self-distillation to learn robust and invariant representations such that we can effectively adapt to a novel data distributions at test-time. (2) We exhaustively evaluate the algorithm over 8 challenging detection and segmentation benchmarks and evaluate its effectiveness compared to previous domain adaptation algorithms and other test-time training methods, and show that the method is 5-10x more data efficient.

2. Related Work

Domain adaptation. In domain adaptation a model is given labeled data from a source data distribution and unlabeled (or few-labeled) data from a target data distribution, and the goal is to learn features such that the model is able to adapt to the novel data distribution. Such algorithms rely on learning invariant representation that learn the structure of the task such that as long as the semantics of the data remains unchanged, the model should be able to generalize to shifts in distribution. Classical domain adaptation algorithms work by kernel mean matching [14], by matching metric spaces for SVMs [41], or minimizing the moments of distribution of a learned representation space between the source and target data [56]. More recent algorithms perform similar minimization techniques using adversarial optimization [55], cyclic losses [20], using coupled generative models [32], or by learning joint representation spaces [34]. The main challenge with such methods is to learn representations that learn the semantics of the image which being invariant to exogenous variables, such as weather conditions or noise [68].

To improve the representation learning over the source and target domains, domain adaptation methods broadly fall into three categories: methods that try to generate more data by training class-conditioned generative models [45, 69], methods that minimize the distance in the representation space by proposing novel domain adaptation objectives [46, 27, 55], or by adding a regularization to the learned space [50, 5, 35]. The current state-of-the art algorithms that perform domain adaptation for object detection utilize decoupling local and global feature alignment [43], maximize discriminability of the learned representations along with the transferability [4], use a combination of adversarial and reconstruction based approaches [23], among others [60, 61, 67, 47]. To be able to learn such transferable features, classical domain adaptation algorithms require access to the data from the target distribution at training time. This limitation significantly hampers the practical use of such models, since the models can only generalize to target distributions that they have encountered during training.

Test-time adaptation. Test-time Training (TTT) was a paradigm that was recently proposed by [52], which trained a model jointly on the task loss (image classification) and an image-rotation prediction task during training [11]. At test-time, the model was adapted to solve the image rotation task of the test images from a different distribution. The paper argues that during training, solving the image rotation task couples the gradients of the rotation prediction task and the task itself, such that performing gradient steps on the novel data over image rotation prediction objective can help with generalization. Test-time entropy minimization (Tent) built upon TTT, by proposing to minimize the entropy of the predictions over the test-data. Tent performed gradient steps to minimize the entropy of the models predictions. Similarly, [37] proposes to maximize the confidence instead of minimizing the entropy of the predictions, using a confidence heuristic. More recently, Tailoring [1] was proposed to directly encode inductive biases using a meta-learning



(a) Stage 1: Adapt the *teacher*.

(b) Stage 2: Adapt the *student*.

Figure 2: Overview of the Test-time Self-Training architecture (TeST). The model consists of a two-stage training process. We first (a) we train a *teacher* network using consistency regularization to learn invariant representations over the novel data distribution. Then (b) we use the trained teacher to distill the predictions onto the student).

scheme at test time.

Lastly, we note that there was recently a concurrent work that also proposes to perform test-time adaptation using a student-teacher framework [57]. The main differences between TeST and [57] are that (1) they propose to use an InfoMax loss [13] to train the teacher as opposed to consistency regularization and FixMatch [51]; (2) they propose a contrastive learning loss to initialize the student that is re-trained from scratch on the target data as opposed to directly updating the available pre-trained student network. As a consequence, (3) they require significantly more examples despite the simpler problem setting of object recognition. Instead, we tackle challenging computer vision tasks such as image segmentation and object detection, using only a small batch of examples (from 64 to 512).

Self-Training and Student-Teacher framework. Student-teacher and self-training algorithms have recently gained popularity due to their ability to learn robust representations by transferring information from a *teacher* to a student *student* using knowledge distillation [19]. Such algorithms have been proposed for semi-supervised learning [54], object detection [53, 62], image classification [3], few shot learning [64], deep metric learning [40], among others. Recent study in [59] shows how self-distillation with appropriate data augmentation techniques can result in state-of-the-art results on the ImageNet classification benchmark.

3. Preliminaries

Test-time Adaptation. In this paper we consider the test-time adaptation setting, where a model is trained on the source data $x_{\mathcal{S}} \sim \mathcal{P}_{\mathcal{S}}$ but then applied to test data $x_{\mathcal{T}} \sim \mathcal{P}_{\mathcal{T}}$ from a potentially different distribution, that is $\mathcal{P}_{\mathcal{S}} \neq \mathcal{P}_{\mathcal{T}}$. The label space is assumed to be the same. During source model training we only have access to pairs from the source domain $(x, y) \sim X_{\mathcal{S}}$ and a model is trained using a cross-entropy loss for semantic segmentation, or a sum of cross-entropy and regression loss for object detection. We denote a model trained only on the source data as $\theta_{\mathcal{S}}$.

Self-Training. In a typical self-training framework, a teacher is trained on the labels from the dataset, while the student is trained on the pseudo-labels from the teacher's predictions on the data (the student has no access to the real labels). The student is typically trained using a knowledge distillation loss from the teacher's predictions [19]; this setup has shown to result in robust predictions for the student. After training, the student is then used on the test-set to perform inference.

4. Test-time Self-Training

We assume that the input to TeST is a model that was already trained on the source data. Unlike [52], the model does not need an auxiliary loss during training on the source data and can be trained in any way. To solve the problem of test-time adaptation, we adopt a student-teacher framework in a two-stage setup. As we do not assume having access to multiple models trained on the source data distribution $P_{\mathcal{S}}$ (e.g. bootstrapping), we copy the source model to create a student $f_{\mathcal{S}}$ and a teacher model $f_{\mathcal{T}}$, thus initially sharing parameters.

In the **first stage** of adaptation, we only train one of the copied models (the teacher) using consistency regularization to learn invariant representations for the test data using an additional randomly initialized two layer predictor network as in Figure 2a. The teacher network adapts to the new domain to produce useful pseudo-labels to the student

in the **second stage**. After the teacher is trained on the test-distribution, we utilize self-training to keep the teacher fix and train the student. The training objective is supervised and uses the test-data and the pseudo-labels from the teacher. This is depicted in Figure 2b. This self-training step transfers the learned invariant representations from the teacher to the predictions of the student. Finally, similar to [58], we add a regularizer to minimize the entropy of the students predictions on the testing distribution which we empirically found to further improve the performance. In the following sections, we will describe each component of the proposed method.

4.1. Stage 1: Training the teacher model

The teacher will be used to produce pseudo-labels for the student training. At this point we have no supervised signal but just samples from the test distribution, so the goal of training is to learn invariant representations which we hope will translate to mearningful pseudo-labels. We implement this by consistency regularization and FixMatch [51] as shown in Figure 2a. We take a *weak* and a *strong* augmentation of the same image from the target distribution to learn representations using extreme examples to enforce stronger invariances. In practice, for the weak augmentations we use simple transformations such as random rotations, translations and crops. As strong augmentation we chose RandAugment [7], which is trained to find the best augmentations for the target task. RandAugment is also a fast alternative to other automatic augmentation variants, and therefore can be applied directly at test time to generate strong augmentations for training the teacher.

After obtaining the features for the strong and weakly augmented images, we use a *predictor* network p_ϕ, which is a two-layer ReLU-MLP network, randomly initialized during the beginning of test-time adaptation following prior work on self-supervised contrastive learning [15]. Using the teacher $f_\mathcal{T}$ and the predictor network p_ϕ, we perform consistency regularization in the feature space to minimize the distance between the encoded representations as shown in Figure 2a. Unlike [51, 49], we minimize the ℓ_2 distance between the strong and the weak embedding on few samples from the target distribution

$$\min_{\theta_\mathcal{T}} \Big|\Big| f_\mathcal{T}\big(strong(x)\big) - p_\phi \circ f_\mathcal{T}\big(weak(x)\big) \Big|\Big|_2^2, \qquad (1)$$

where $strong(\cdot)$ and $weak(\cdot)$ are strong and weak augmentation policies respectively and $\theta_\mathcal{T}$ are the teacher parameters. Learning invariant representations is key to adaptation to the novel test distribution. As our main focus are object detection and segmentation, we remark that stage 1 only updates the feature encoder of the teacher networks.

Input: Teacher steps M; student steps N;
Source model f_{source}; batch of test data $x_\mathcal{T}^m$
Output: Student model parameters $\theta_\mathcal{S}$.

$f_\mathcal{T} \leftarrow f_{source}$; $f_\mathcal{S} \leftarrow f_{source}$; $i, j \leftarrow 0$
while $i < M$ **do**
  $\mathcal{L}_\mathcal{T} \leftarrow \Big|\Big| f_\mathcal{T}\big(strong(x)\big) - p_\phi \circ f_\mathcal{T}\big(weak(x)\big) \Big|\Big|_2^2$
  $\theta_\mathcal{T} \leftarrow \theta_\mathcal{T} - \alpha \nabla \mathcal{L}_\mathcal{T}$
  $i \leftarrow i + 1$
end while
while $j < N$ **do**
  $\tilde{y} \leftarrow f_\mathcal{T}(x_\mathcal{T})$
  $\mathcal{L}_\mathcal{S} \leftarrow \mathcal{L}_{KD}(f_\mathcal{S}(x_\mathcal{T}), \tilde{y}) + \lambda \mathcal{H}(f_\mathcal{S}(x_\mathcal{T}))$
  $\theta_\mathcal{S} \leftarrow \theta_\mathcal{S} - \alpha \nabla \mathcal{L}_\mathcal{S}$
  $j \leftarrow j + 1$
end while

4.2. Stage 2: Training the student model

After training the teacher model, we discard the predictor network p_ϕ since it is not required to obtain the task predictions. Using the invariant features from the teacher model, we then train the student using pseudo-labels from the teacher, implicitly distilling information about the target distribution as shown in Figure 2b. More specifically, given a batch of test images, we use the teacher model to generate pseudo-labels $\tilde{y}$ and use a knowledge distillation objective to fine-tune the student's predictions. Using knowledge distillation [19] from the teacher to the student results in robust representations learned by the student as the representations learned by the teacher are implicitly learned by the student. The per-image x (on the target distribution) training objective reads

$$\min_{\theta_\mathcal{S}} \mathcal{L}_{KD}(f_\mathcal{S}(x), f_\mathcal{T}(\tilde{y}|x)) \qquad (2)$$

where $\mathcal{L}_{KD}$ is the knowledge distillation loss, $f_\mathcal{S}$ is the student network parameterized by $\theta_\mathcal{S}$. We note that the loss is only minimized over the students parameters $\theta_\mathcal{S}$, and the teacher $f_\mathcal{T}$ is kept fixed. The knowledge distillation loss used for a categorical variable (such as bounding box classification) is the KL-Divergence between the two probability distributions, and for a continuous variable (such as bounding box regression) is the L2 distance between the pseudo-labels and the student labels.

To improve the confidence of the student predictions, we add an entropy minimization term over the probability distribution of the class predictions, as confidence has been linked to performance [58]. The entropy minimization term is added as a regularization to the student training. Adding a weighted entropy term to the objective in Equation 2, the full student loss is thus:

$$\min_{\theta_\mathcal{S}} \mathcal{L}_{KD}(f_\mathcal{S}(x), f_\mathcal{T}(\tilde{y}|x)) + \lambda \mathcal{H}(f_\mathcal{S}(x)), \qquad (3)$$

where λ is a hyperparameter that weights the entropy term $\mathcal{H}$ sharpening the confidence of the class prediction.

4.3. Concluding remarks

To summarize, we adapt a pre-trained source model $f_{\mathcal{S}}$ in two stages to the novel test distribution $\mathcal{P}_{\mathcal{T}}$: first we train the teacher by minimizing the consistency regularization loss using strong and weak augmentations of the target images, and secondly we use the trained teacher to provide pseudo-labels for the student to train on. The student is then optimized using gradient descent over the weighted knowledge distillation and entropy minimization loss, as in Eqn. 3, see Section 5.1 for experimental details. The joint training algorithm can be found in Algorithm 1. This procedure is generic and can be used for many computer vision domain adaptation task. In the following we test the performance of TeST on object detection and image segmentation tasks.

5. Experiments

In this section, we seek to answer the following questions:

- Can TeST yield better domain adaptation performance on large-scale object detection tasks than baseline test-time adaptation methods? How does it compare against domain adaptation methods where test data is available during training time?
- Can TeST yield better domain adaptation performance on further computer vision tasks, such as image segmentation?
- Is TeST robust to our assumption of distribution shift and performs better representation learning even if the testing distribution is the same as the training distribution?

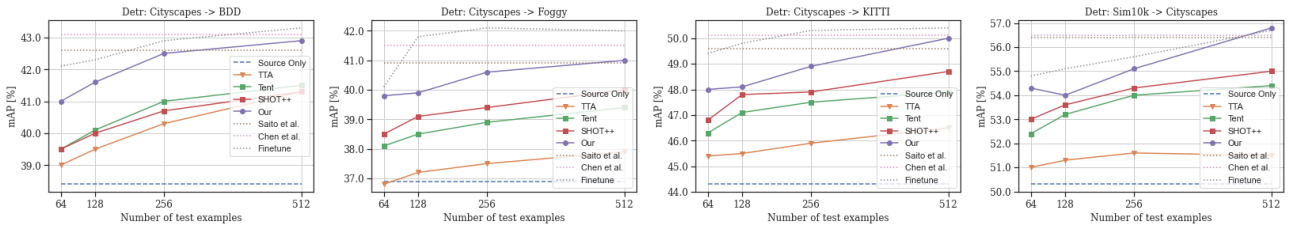
5.1. Experimental setup

Test data budget: In test-time adaptation, the model is able to adapt to a certain *budget* n of images available during test-time to adapt to the novel data distribution. To thoroughly examine our proposed solution, we test over a range of values for $n \in \{64, 128, 256, 512\}$.

Baselines: We consider a range of baselines. **Source Only**: refers to training on source data $X_{\mathcal{S}}$ only and no further adaptation. In addition, we use two recent test-time adaptation models as baselines for all object detection experiments: **Test-Time Training** (TTT) which proposes to solve an image rotation prediction task at training time and test time to adapt to novel distributions [52]. **Tent**: which proposes to minimize the entropy of the test predictions as an unsupervised task at test-time [58]. Furthermore, we also show results with the model being trained with access to the test-distribution without labels during training time as considered in unsupervised domain adaptation setting. **SHOT**: which proposes to use a self-supervised objective to maximize the alignment of the features between the original source data and the novel target distribution [29]. Namely, we use two recently proposed unsupervised domain adaptation algorithms **Saito et al.** [43] and **Chen et al.** [4]. Finally, we also compare our results with a **Finetuning** oracle, where we consider the performance of the trained model on the test-set if it had access to n labeled samples. Although it is possible to improve the *fine-tune* results using more recent transfer learning algorithms [16, 25, 28], we update all parameters of the network trained on the source domain using the ground-truth label information as opposed to the pseudo-labels in the knowledge distillation loss of Equation 3 (without the entropy regularization).

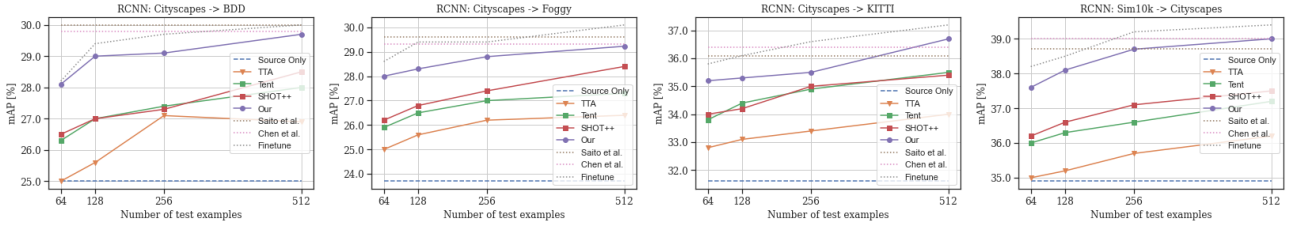
Architectures: For the object detection experiments, we use two architectures: Faster RCNN [38] and Deformable Detection Transformer [70] (DeTR). We train them on the source datasets using the default hyperparameters mentioned in the original papers. For image segmentation, we train a Dilated Residual Network [65] on the source data again using default hyperparameters.

Datasets: We consider a variety of self-driving object detection domain adaptation challenges, namely: Cityscapes [6] $\rightarrow$ BDD [66], Cityscapes $\rightarrow$ Foggy Cityscapes [44], Cityscapes $\rightarrow$ KITTI [9] and Sim10k [22] $\rightarrow$ Cityscapes. This set of experiments measure a variety of distribution shifts, such as weather, location and sim-2-real. Additionally, we also generate OOD splits from MS-COCO [31] by extracting the features using a trained ResNet-101 model [18], and then generating k-clusters of the embedding to separate the data along some unknown visual or semantic boundaries. The motivation is twofold. First, we want to test the adaptation algorithms on *common objects* as in the ones in MS-COCO. Second, generating k-clusters and varying k we can control the severity of the distribution shift between training and test time and evaluate how this affects performance of the tested approaches. We hold out one cluster and use it as a *OOD* target distribution, training the model on the $k-1$ clusters. Smaller k corresponds to more severe shift as some object classes may even be missing from the source training distribution. However, manually checking the label distribution of the clusters, confirmed that at least a few examples of each of the classes are present in each cluster. More details regarding all test-settings and datasets can be found in Appendix A.



(a) Cityscapes → BDD100k (b) Cityscapes → Foggy (c) Cityscapes → KITTI (d) Sim10k → Cityscapes

Figure 3: Results on object detection with self-driving datasets where the base detector is a Deformable Detection Transformer is used [70]. We see that we are able to significantly outperform other test-time adaptation models (solid lines), while being comparable to domain adaptation models [4] (brown dotted line) and [43] (violet dotted line) which require 5-10x more data from the target data distribution. TeST is also comparable to the **Finetuning** baseline, which directly finetunes on the budget of ground-truth labels on the test distribution, instead of using an unsupervised objective.



(a) Cityscapes → BDD100k (b) Cityscapes → Foggy (c) Cityscapes → KITTI (d) Sim10k → Cityscapes

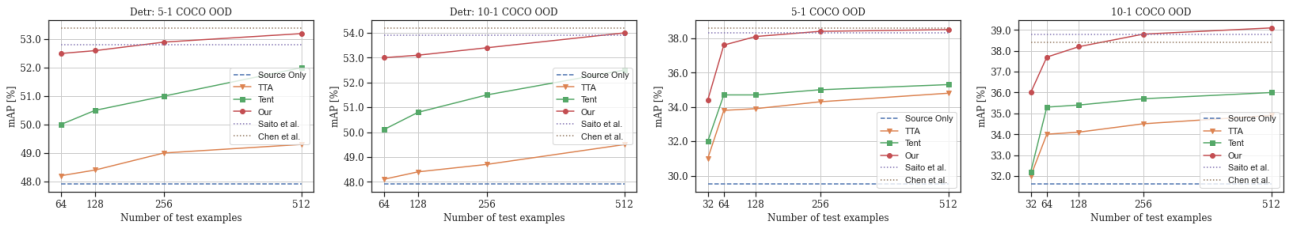
Figure 4: Results on object detection with self-driving datasets with a Faster RCNN base object detector [38]. TeST significantly outperforms the each of the test-time adaptation baselines (solid lines), and is comparable and often better than the domain adaptation baselines [4] (brown dotted line) and [43] (violet dotted line). TeST is also able to obtain similar performance to the *oracle* baseline, which finetunes on the labeled data points, instead of using an unsupervised objective.

Hyperparameters: To train TeST, we use an Adam optimizer [24] with a learning rate of 3×10^{-4} for both the student and teacher, and perform 10 epochs over the n-budget for each of the stages. As mentioned previously, to generate *strong* augmentations, we use a RandAugment policy [7]; to generate *weak* augmentations, we perform rotations from $[-10°, 10°]$, and random crops of the original image. For all experiments, we use an entropy weight $\lambda = 0.25$.

5.2. Results

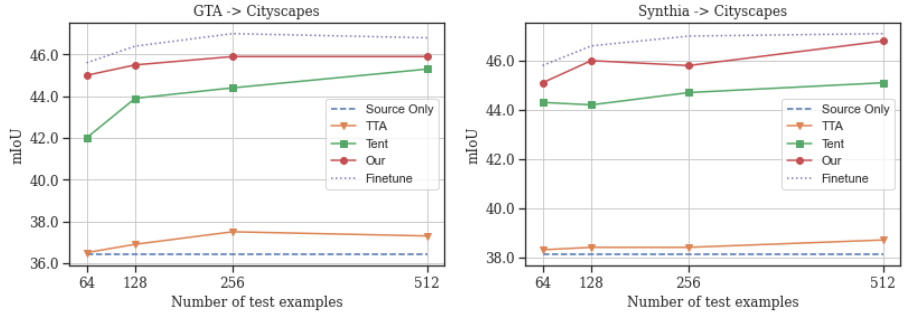
Self-Driving Object Detection Benchmarks We test over a variety of benchmark self-driving object detection datasets and their domain adaptation variants which include change in weather (Cityscapes → Foggy Cityscapes), change in location (Cityscapes → KITTI and Cityscapes → BDD) and from simulated to real data (Sim10k → Cityscapes). The results for Deformable DeTR are shown in Figure 3, and the results for FasterRCNN can be found in Figure 4. In both cases, we see that TeST is able to significantly outperform both the test-time adaptation baselines and the model that was only trained on source data (*Source Only* in the plots). Interestingly, we note that TeST is able to achieve performance comparable to unsupervised domain adaptation algorithms from Chen et al.[4] and Saito et al. [43] despite only using between 256 and 512 test-images. Both of the domain adaptation algorithms were trained with the whole training dataset of the target tasks (see Appendix A for further details). This results in TeST performing comparably to such algorithms, while being 5-10x more data efficient, as TeST only requires 256-512 unlabeled images to obtain similar performance. This suggests that instead of assuming that all source images are available at training time, like in [43, 4], test-time adaptation provides a data-efficient alternative. We also note that TeST performs comparably at times to the oracle *finetuning* baseline where the model is finetuned on n labeled examples, as opposed to having to perform unsupervised adaptation at test-time.

Object Detection on MS-COCO MS-COCO is a versatile benchmark dataset for several computer vision tasks including object detection. To test domain adaptation on MS-COCO we split the data in k clusters based on the trained embeddings of a ResNet-101 model. Training is then done on $k-1$ clusters, the held-out cluster is used for testing. This clustering technique separates creates an OOD testset with an unknown domain shift. Improving the perfor-



(a) DeTR: 4 train - 1 test (b) DeTR: 9 train - 1 test (c) RCNN: 4 train - 1 test (d) RCNN: 9 train - 1 test

Figure 5: Results on object detection using the MS-COCO dataset with Deformable DeTR and Faster RCNN. We cluster the MS-COCO dataset using the embeddings of a pretrained ResNet feature extractor into k-clusters. We then randomly select one cluster to be the out-of-distribution test set, and use the remaining $k-1$ clusters for training. This way we are able to evaluate domain adaptation algorithms on *unknown* distribution shifts, as the clusters are separated over an unknown semantic boundary. We continue to see that TeST is able to outperform the test-time adaptation baselines (solid lines) and is competitive with unsupervised domain adaptation models (dotted lines) which are trained with data from all k-clusters.



(a) GTA $\rightarrow$ Cityscapes (b) Synthia $\rightarrow$ Cityscapes

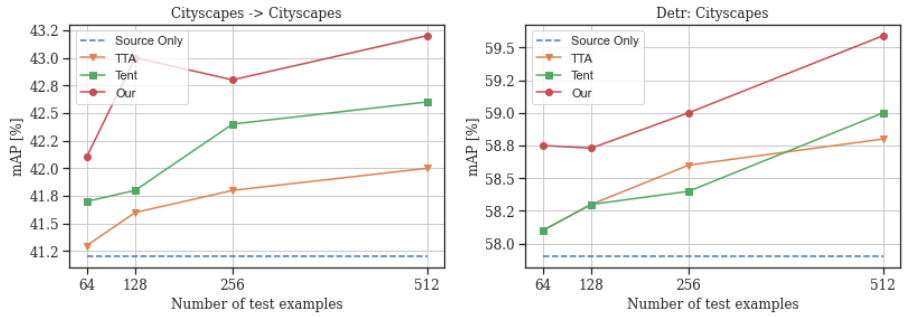
Figure 6: Results on image segmentation with DRN over two challenging Sim2Real driving tasks. We compare TeST to 2 test-time adaptation algorithms, and directly finetuning, and see that TeST is comparable to finetuning on the labels directly.

mances on unknown domain shifts is important to ensure that the models did not overfit to commonly tested distribution shifts, such as weather, location, an sim-2-real. We detail the dataset splitting further in Appendix A. In practice, we vary the total number of clusters, $k \in \{5, 10\}$, which means 4 or 9 clusters are available for training, respectively, while 1 is held out for testing. The fewer clusters, the larger the expected distribution shift between the training and test distributions. As before, we train a Faster-RCNN object detector [38] and a Deformable DeTR [70] on the training clusters, and then evaluate the chosen test-time adaptation baselines, and TeST on the testing cluster.

Full results on MS-COCO are present in Figure 5. Similar as before, we see considerable improvements compared to the test-time baselines, while again achieving comparable performance to the domain adaptation algorithms despite using significantly less data. As expected, we observe that the performance is overall slightly worse for more severe shifts (fewer clusters). However, this has no effects on the conclusions we can draw, confirming the trends we observed on the other datasets. This further suggests that TeST is able to get empirical gains across different domains (self-driving datasets and common objects) across two popular object detectors and severity of the shift.

Semantic Segmentation for Self-Driving To test the generality of TeST, it is important for the algorithm to be agnostic to the type of task considered. Towards this, we further perform experiments with semantic segmentation on self-driving datasets. We consider the same test-time setting as before, and use the popular Sim2Real benchmarks of GTA $\rightarrow$ Cityscapes and Synthia $\rightarrow$ Cityscapes [39], where the model is trained on simulation data, and tested on the natural-image Cityscapes dataset [6]. The model is trained using a Dilated Residual Network [65] using the default hyperparameters as suggested in the original paper.

The results are shown in Figure 6. Again, we see that TeST is able to outperform the baseline algorithms by a considerable margin, which suggests that the proposed method is indeed task-agnostic and applies beyond detection only.. Interestingly, we see that Tent [58] significantly outperforms the Test-Time Training baseline (TTT) which also



(a) Faster-RCNN (b) Deformable DeTR

Figure 7: Results on object detection for Cityscapes (no OOD) using Faster RCNN (left) and Deformable DeTR (right). The results suggest that even when there is no distribution shift between the training and the testing distribution, TeST is able to outperform other test-time adaptation baselines. This further shows that TeST is robust to any or no distribution shift.

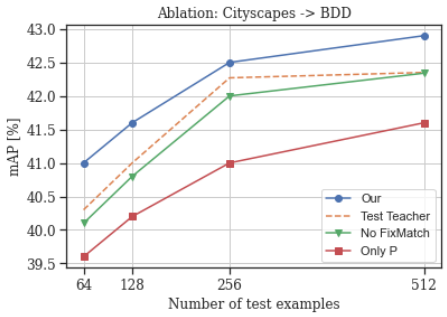


Figure 8: Ablation study over different possible variants of TeST on the benchmark Cityscapes → BDD domain adaptation challenge.

shows the importance of entropy minimization at test-time, which is also a component of TeST.

Investigating No Distribution Shift TeST assumes that the target distribution is different than the source distribution. This raises the natural question of what happens whenever there is actually **no** distribution shift between the training and the testing distribution. Our goal is to understand whether TeST is robust to violation of this assumption as in practical scenarios it may be difficult to know for certain that the distribution has changed. Prior work found that self-distillation improves performance on the training dataset [8]. While our approach was designed to cope with distribution shifts, we would still hope to see gains even if the distribution did not change. For this, we use the original test set of the source dataset itself. Similar to before, we use the Cityscapes dataset [6] for training, and evaluate on the held-out validation set.

The results for a Faster-RCNN and Deformable DeTR detectors are in Figure 7 and continue to show the broad applicability of TeST: even if the distribution did not change, we are able to significantly improve performance.

Ablation Study Finally, we present different variations of TeST and report their performance on the Cityscapes → BDD object detection benchmark using a Deformable DeTR in Figure 8. We observe the largest performance decrease if we only perform knowledge distillation on the probability output p of the bounding box i.e. no bounding box regression (denoted as "Only p" in Figure 8). Similarly, we can see the benefit of learning invariant representations in the teacher network using FixMatch and the consistency regularization as opposed to simply doing self-distillation (No FixMatch). Finally, we also note that it is possible to use the Teacher directly ("Test Teacher" in Figure 8), however, using knowledge distillation clearly helps to learn a robust student. Overall, each component of the method yields a positive contribution to the final performance. This further suggests the TeST builds upon several components that work well together, to perform better representation learning through self-training.

6. Conclusion

In this paper, we propose TeST: a test-time adaptation technique that uses self-training in a student-teacher framework to overcome distribution shifts at test-time. The key ingredients are (1) learning invariant and robust representations of the test distribution, and (2) distilling the predictions to the student model. TeST consists of two-stage, a teacher training through consistency regularization followed by knowledge distillation and entropy minimization to train a student model. Overall, TeST significantly outperforms test-time adaption baselines, and is comparable to unsupervised domain adaptation techniques that require 5-10x more data from the target distribution during training.

References

[1] Ferran Alet, Maria Bauza, Kenji Kawaguchi, Nurullah Giray Kuru, Tomas Lozano-Perez, and Leslie Pack Kaelbling. Tailoring: encoding inductive biases by optimizing unsupervised objectives at prediction time. *arXiv preprint arXiv:2009.10623*, 2020.

[2] Shuang Ao, Xiang Li, and Charles Ling. Fast generalized distillation for semi-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[3] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov. Knowledge distillation: A good teacher is patient and consistent. *arXiv preprint arXiv:2106.05237*, 2021.

[4] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8869–8878, 2020.

[5] Safa Cicek and Stefano Soatto. Unsupervised domain adaptation via regularized conditional alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1416–1425, 2019.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020.

[8] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[10] Robert Geirhos, Carlos R Medina Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *arXiv preprint arXiv:1808.08750*, 2018.

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[12] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[13] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Nueral Information Processing Systems*, 2010.

[14] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[16] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4805–4814, 2019.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[20] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[21] Pin Jiang, Aming Wu, Yahong Han, Yunfeng Shao, Meiyu Qi, and Bingshuai Li. Bidirectional adversarial training for semi-supervised domain adaptation. In *IJCAI*, pages 934–940, 2020.

[22] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

[23] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Zhi Kou, Kaichao You, Mingsheng Long, and Jianmin Wang. Stochastic normalization. *Advances in Neural Information Processing Systems*, 33, 2020.

[26] Fabian Kuppers, Jan Kronenberger, Amirhossein Shantia, and Anselm Haselhoff. Multivariate confidence calibration for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 326–327, 2020.

[27] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019.

[28] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, Zeyu Chen, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. *arXiv preprint arXiv:1901.09229*, 2019.

[29] Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[32] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. *Advances in neural information processing systems*, 29:469–477, 2016.

[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[34] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, pages 2208–2217. PMLR, 2017.

[35] Xudong Mao, Yun Ma, Zhenguo Yang, Yangbin Chen, and Qing Li. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*, 2019.

[36] Timo Milbich, Karsten Roth, Samarth Sinha, Ludwig Schmidt, Marzyeh Ghassemi, and Björn Ommer. Characterizing generalization under out-of-distribution shifts in deep metric learning. *arXiv preprint arXiv:2107.09562*, 2021.

[37] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021.

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[39] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[40] Karsten Roth, Timo Milbich, Bjorn Ommer, Joseph Paul Cohen, and Marzyeh Ghassemi. Simultaneous similarity-based self-distillation for deep metric learning. In *International Conference on Machine Learning*, pages 9095–9106. PMLR, 2021.

[41] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

[42] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019.

[43] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

[44] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[45] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018.

[46] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[47] Zhiqiang Shen, Harsh Maheshwari, Weichen Yao, and Marios Savvides. Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *arXiv preprint arXiv:1911.02559*, 2019.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[49] Samarth Sinha and Adji B Dieng. Consistency regularization for variational auto-encoders. *arXiv preprint arXiv:2105.14859*, 2021.

[50] Samarth Sinha, Karsten Roth, Anirudh Goyal, Marzyeh Ghassemi, Hugo Larochelle, and Animesh Garg. Uniform priors for data-efficient transfer. *arXiv preprint arXiv:2006.16524*, 2020.

[51] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[52] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[53] Yihe Tang, Weifeng Chen, Yijun Luo, and Yuting Zhang. Humble teachers teach better students for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3132–3141, 2021.

[54] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets im-

prove semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[55] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[56] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[57] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *arXiv preprint arXiv:2109.01087*, 2021.

[58] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[59] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[60] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020.

[61] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12355–12364, 2020.

[62] Mengde Xu, Zheng Zhang, Han Hu, Jianfeng Wang, Lijuan Wang, Fangyun Wei, Xiang Bai, and Zicheng Liu. End-to-end semi-supervised object detection with soft teacher. *arXiv preprint arXiv:2106.09018*, 2021.

[63] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2142–2150, 2015.

[64] Han-Jia Ye, Lu Ming, De-Chuan Zhan, and Wei-Lun Chao. Few-shot learning with a strong teacher. *arXiv preprint arXiv:2107.00197*, 2021.

[65] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 472–480, 2017.

[66] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

[67] Hui Zhang, Yonglin Tian, Kunfeng Wang, Haibo He, and Fei-Yue Wang. Synthetic-to-real domain adaptation for object instance segmentation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019.

[68] Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

[69] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *European Conference on Computer Vision*, pages 561–578. Springer, 2020.

[70] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

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.

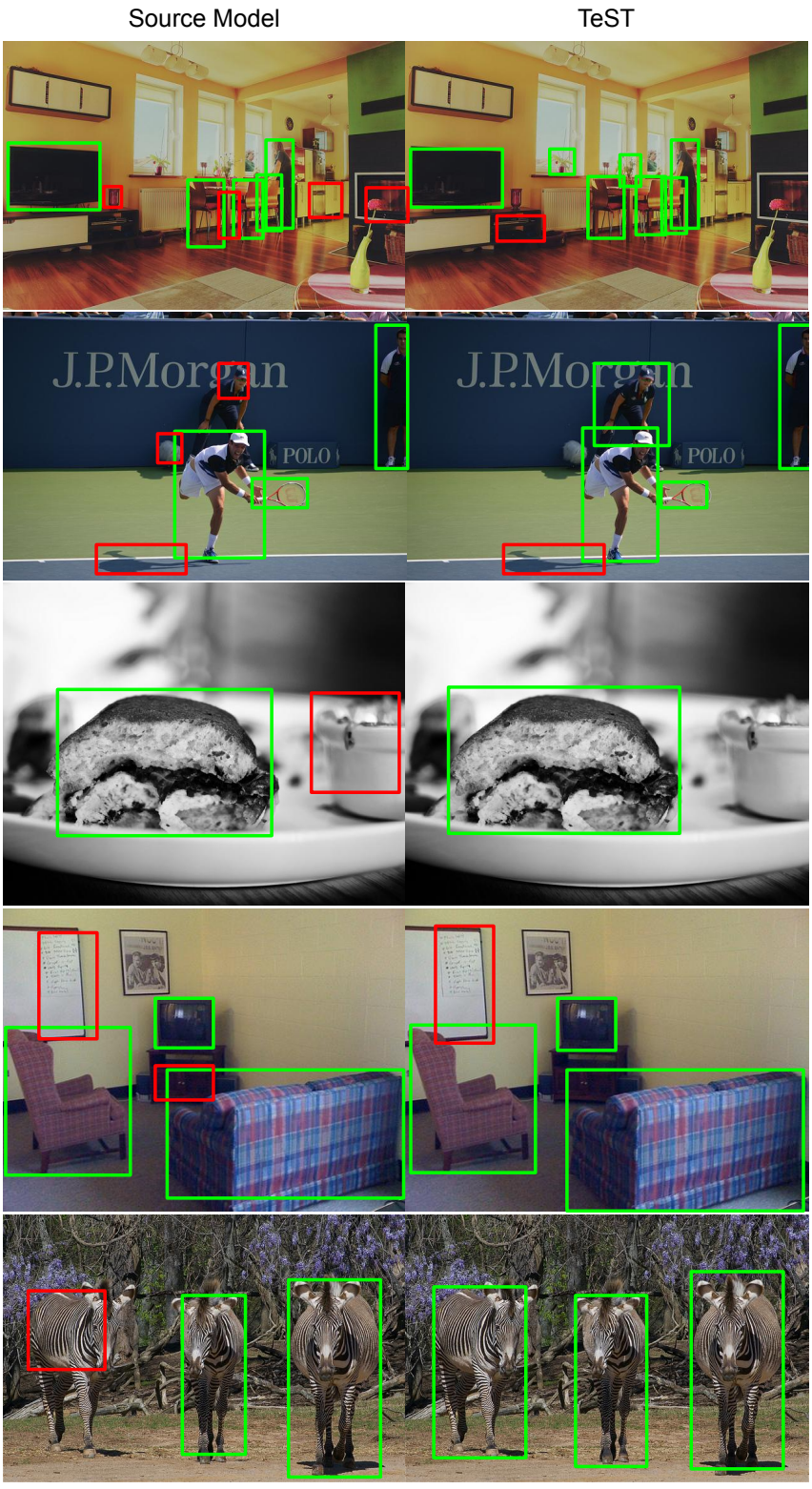


Figure 10: Qualitative results from MS-COCO with a Faster RCNN Object Detector [38]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST have 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.

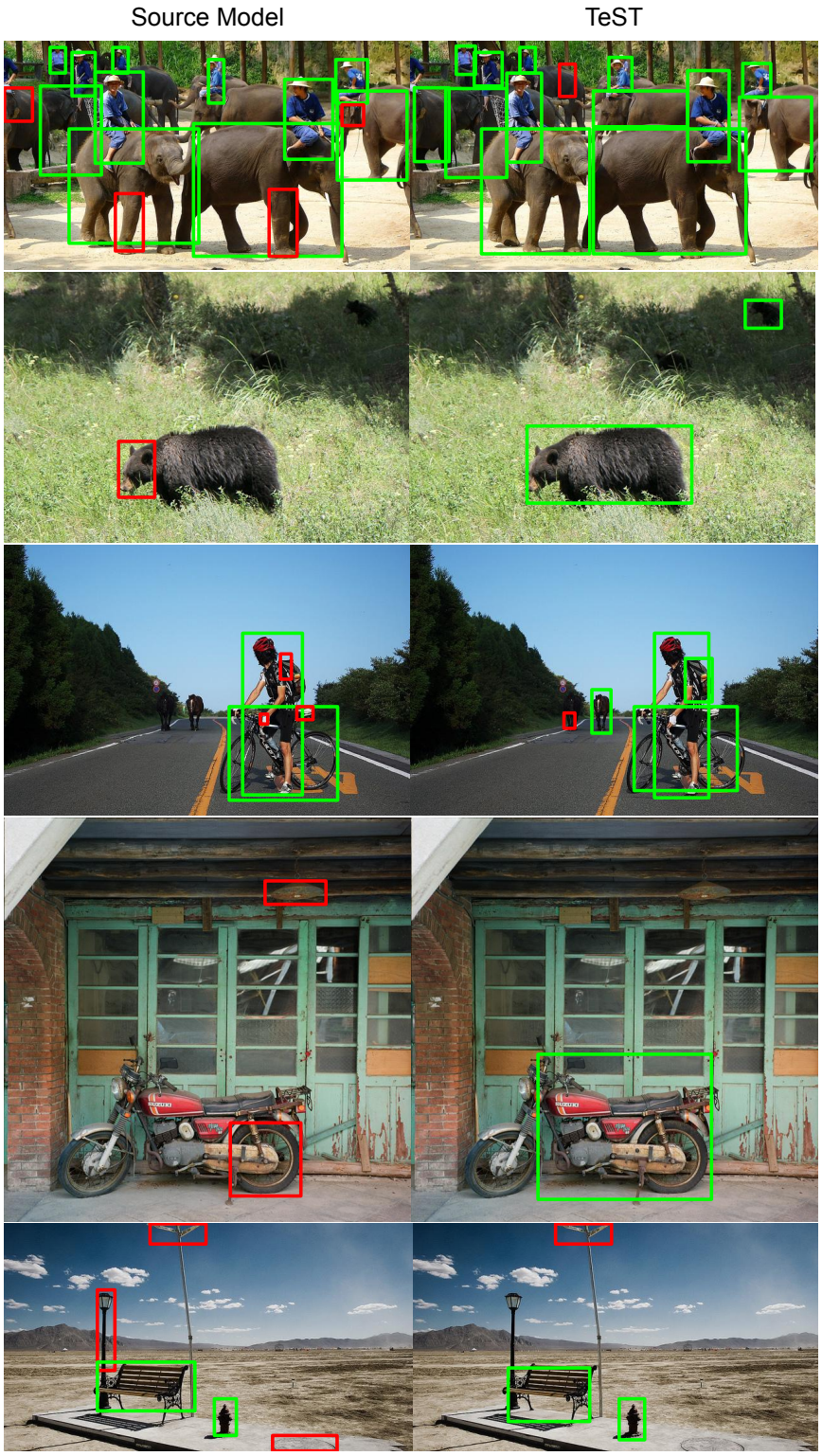


Figure 12: Qualitative results from MS-COCO with a Deformable DeTR Object Detector [70]. True positives are shown in green rectangles and False positives are shown in red rectangles. We note that: **all images are chosen at random without any cherry-picking.** We see that the models trained with TeST 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.

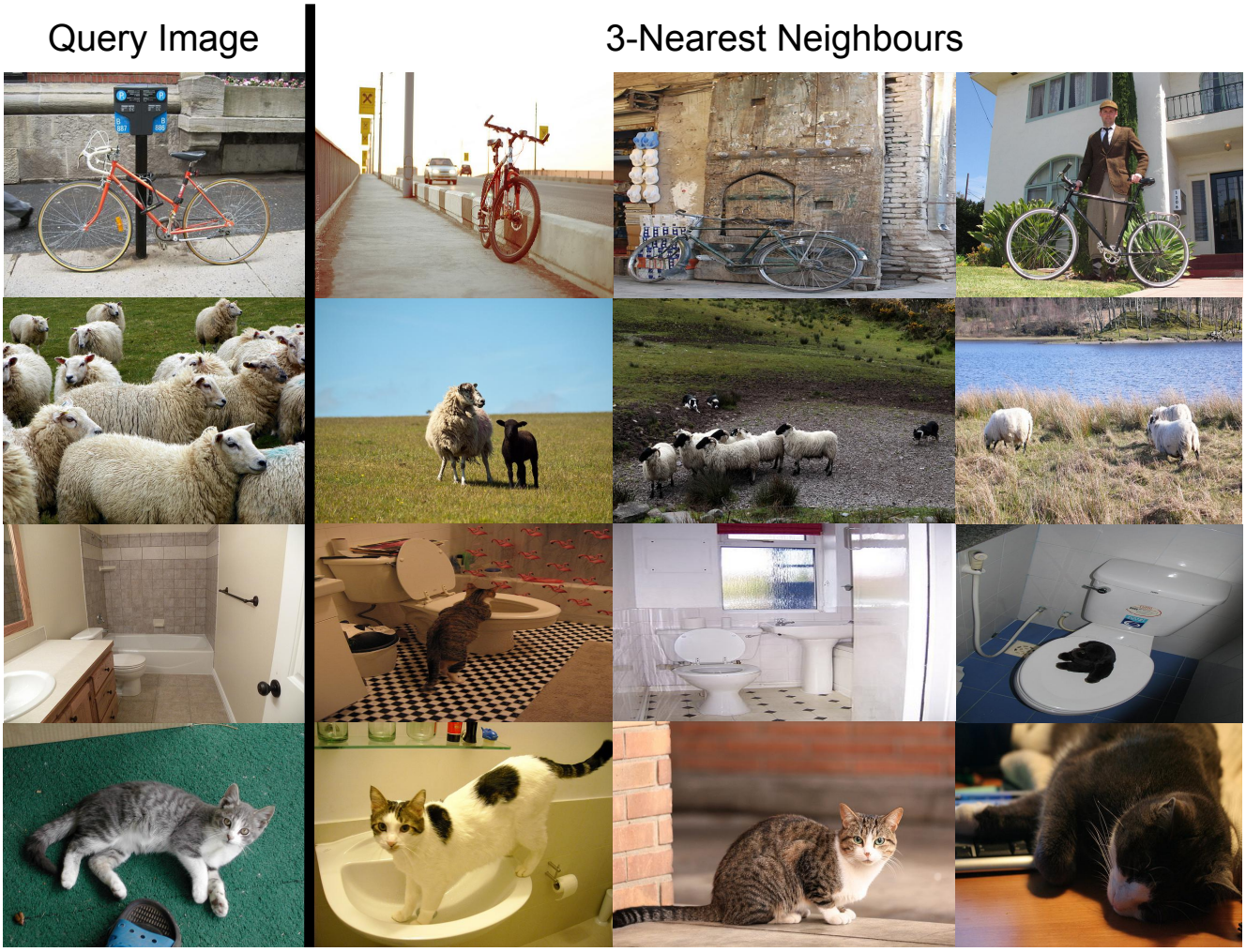


Figure 13: 3-Nearest Neighbours in the embedding space for the feature extractor of a Faster-RCNN detector [38], after it has been trained using TeST on the COCO dataset. We see that the model is able to learn semantically meaningful representations and the nearest neighbours to the query image are semantically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