Robust Object Detection via Instance-Level Temporal Cycle Confusion

Xin Wang[1] Thomas E. Huang[2*] Benlin Liu[3*]
Fisher Yu[2] Xiaolong Wang[4] Joseph E. Gonzalez[5] Trevor Darrell[5]

[1]Microsoft Research [2]ETH Zürich [3]University of Washington [4]UC San Diego [5]UC Berkeley

Abstract

Building reliable object detectors that are robust to domain shifts, such as various changes in context, viewpoint, and object appearances, is critical for real-world applications. In this work, we study the effectiveness of auxiliary self-supervised tasks to improve the out-of-distribution generalization of object detectors. Inspired by the principle of maximum entropy, we introduce a novel self-supervised task, instance-level temporal cycle confusion (CycConf), which operates on the region features of the object detectors. For each object, the task is to find the most different object proposals in the adjacent frame in a video and then cycle back to itself for self-supervision. CycConf encourages the object detector to explore invariant structures across instances under various motions, which leads to improved model robustness in unseen domains at test time. We observe consistent out-of-domain performance improvements when training object detectors in tandem with self-supervised tasks on various domain adaptation benchmarks with static images (Cityscapes, Foggy Cityscapes, Sim10K) and large-scale video datasets (BDD100K and Waymo open data)[1].

1. Introduction

Object detection has achieved remarkable performance on in-domain data [2, 32]. However, contemporary visual perception models still suffer significant performance degradation under domain shifts, raising concerns for safety critical applications such as autonomous driving [5, 23].

Prior works [1, 3, 17, 21, 33, 51] have designed domain adaptive object detectors, which align the unlabeled target domain data and labeled source domain data in the feature space to tackle the distribution shift problem. These approaches perform well when they have access to excessive unlabeled data in the target domain. More recently, another line of work improves model robustness to domain shift [22, 48], image corruption and distortion [13] through pre-training [14], self-supervision [15], and data augmentation [16]. These studies are largely conducted on image classification, and the effectiveness is unknown for structural prediction tasks like object detection.

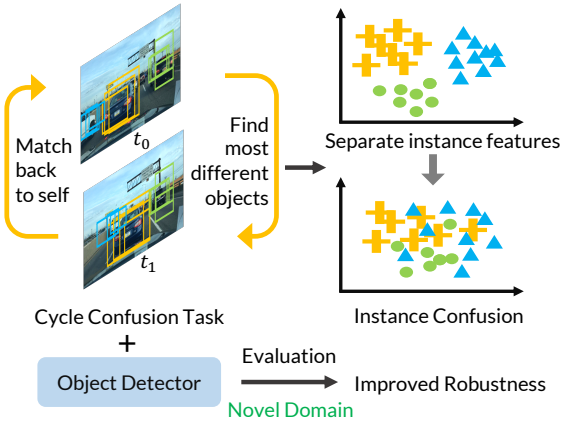


Figure 1: We introduce a self-supervised task, instance-level temporal cycle confusion (CycConf), which mixes up instance features to encourage learning invariant structures across instances. The object detector trained in tandem with the auxiliary self-supervised task is more robust to domain shifts.

In this work, we revisit the idea of auxiliary self-supervised tasks to improve out-of-domain generalization of object detectors. Through empirical studies, we find the widely used self-supervised tasks such as image rotation [10] and Jigsaw [26], used in tandem with the fully supervised object detector in the source domain, can consistently improve the object detector's out-of-domain performance (e.g., evaluating on different datasets or different scenes) without using target data. Surprisingly, we also find that jointly training the object detector and the rotation task outperforms the complicated feature alignment approaches by a large margin on a range of unsupervised domain adaptation (UDA) benchmarks, where the test domain is known during training. These findings indicate the usage of auxiliary self-supervised tasks can be a general solution to improve model robustness under various assumptions about the amount of unlabeled target data available.

*Equal contribution. The authors are listed in alphabetical order.
[1]The models are released at https://xinw.ai/cyc-conf

While the findings are inspiring, we take a step further to ask, what would be a good self-supervised task for out of domain object detection? Here we introduce a new self-supervised task, instance-level cycle confusion (CycConf), which operates on the region features of the object detectors as shown in Figure 1. For a pair of frames in a video sequence, the CycConf task is to find the *most different* objects through time in the frames. Inspired by the principle of maximum entropy [18, 19], CycConf mixes up the instance features to encourage the model to explore invariant structures across instances which may be under various motion, viewpoint and lightning conditions. In contrast to object tracking, which finds identical objects through time, CycConf encourages cross-instance matching which increases confusion among instances and encourages the object detector to explore the latent structures of the instances that are invariant to changing environments. Therefore, the object detector trained in tandem with CycConf has improved robustness to domain shifts.

To evaluate the effectiveness of our new self-supervised task, we construct a benchmark of out-of-domain generalization for object detection using the video datasets BDD100K [48] and Waymo Open data [39], which are the largest contemporary driving video datasets in the open source community. The datasets contain various object scales and diverse scenes, which is a good testbed for new model designs. In this benchmark, we consider various out-of-domain scenarios (e.g., generalization across different time of day, camera views and datasets). The proposed CycConf task improves the baseline model by 2 to 5 points in average precision (AP50), outperforming other self-supervised tasks when evaluating the object detector in unseen domains. Our contributions can be summarized as follows.

- We show the adoption of auxiliary self-supervised tasks is a general solution to improve the robustness of object detectors to domain shifts under various assumptions.
- We introduce instance-level cycle confusion (CycConf), a novel self-supervised task on instance features, which improves the object detectors' robustness.
- The proposed approach achieves state-of-art performances on a range of domain adaptation benchmarks. We additionally introduce an out-of-domain generalization benchmark for object detection using large-scale driving videos.

2. Related Work

Our work is in line with the model robustness and domain adaptation literature with an emphasis on the object detection task. The design of CycConf is conceptually related to the principle of maximum entropy and connects to other self-supervised tasks in the literature.

Domain adaptation and robustness. Generalization under domain shifts is a core problem in machine learning and computer vision. Many [1, 3, 17, 21, 33, 38, 46, 47, 50, 51] have designed domain adaptive object and used the unlabeled target domain data through feature alignment. Sun *et al*. [40] use an auxiliary rotation task to leverage the unlabeled target domain data for semantic segmentation. However, accessing massive target data is not feasible in many real-world applications [5, 23]. In another line, some works [9, 13, 31, 36] focus on testing the robustness of image classification models on out-of-domain data at inference, seeking to improve the model robustness to domain shfits [22, 48], image corruption and distortion [13] through pre-training [14], data augmentation [16] and self-supervision [15].

Both Hendrycks *et al*. [15] and Sun *et al*. [40] show that the adoption of an auxiliary rotation task improves model robustness to image corruption and domain shifts. However, they do not touch on the structural prediction tasks like object detection and do not discuss the choices of self-supervised tasks under various assumptions (e.g., the availability of test domain data). In this work, we bridge the gap between the unsupervised domain adaptation and model robustness literature and show that using auxiliary self-supervised tasks is a general solution to improve the out-of-domain performance in different situations. We emphasize on the object detection task and design a new instance-level self-supervised task that better suits object detectors.

Maximum entropy. The principle of maximum entropy was founded by E. T. Jaynes in 1950s [18, 19] for statistical mechanics and information theory, which indicates that the probability distribution with the highest entropy is the one that best represents the current state of knowledge in the context of precisely stated prior data. Maximum entropy reinforcement learning (MaxEnt RL) [30, 41, 52] has a set of robust reinforcement learning algorithms built upon such concept. Eysenbach and Levin [8] recently provide a theoretical justification of the robustness of MaxEnt RL under various environments.

The design of CycConf is conceptually related to the principle of maximum entropy, which increases the entropy of the matching prediction probability distribution by matching the most different objects across frames. From the adversary perspective, CycConf makes it challenging to distinguish the instance identities and encourages the object detector to explore the invariant structure among instances, which may be under various motion, viewpoint or lightning conditions.

Self-supervised learning. In self-supervised learning, a pretext task is usually designed to provide auxiliary signals to train neural networks [6, 10, 26, 29, 42, 49]. For example, the image rotation task [10] applies a rotation transformation to the input image and requires the network to estimate what is the applied transformation. Besides creating pretext tasks in images, researchers have extended self-supervised learning with video data [7, 27, 35, 43, 44]. Dwibedi *et al*. [7] adopt temporal cycle consistency to provide super-

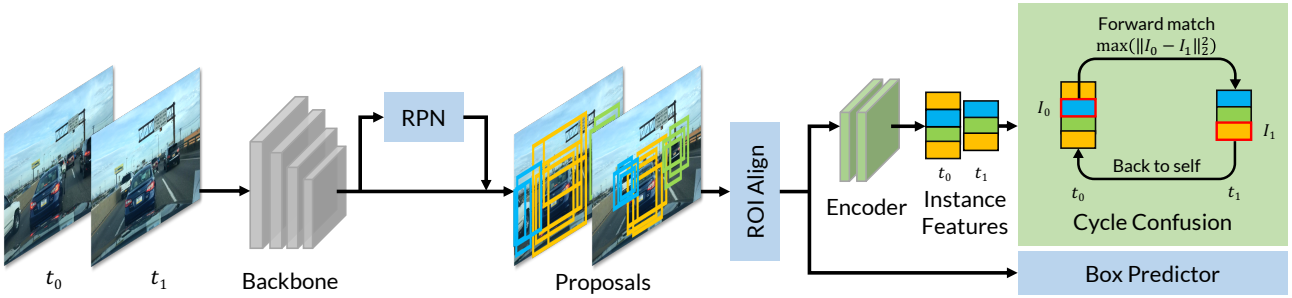


Figure 2: Overall model architecture. A two-stage object detector is trained in tandem with the instance-level cycle confusion (CycConf) task. For a pair of images, we transform the region features produced by the region proposal network (RPN) with two convolutional layers (encoder) and then use the transformed instance features as input to the cycle confusion task shown in **green**. In the forward matching from t_0 to t_1, the matching object I_1 is defined as the object with maximum distance to the object I_0 in t_0 (a.k.a. the most different object). The matching object in t_1 cycles back to the original object in t_0 for self supervision.

vision for learning frame-level correspondence across multiple videos. Wang *et al*. [44] introduce a self-supervised method for learning visual correspondence through cycle-consistency of time. The core idea in these approaches is to enable matching frames using nearest neighbors in the learned per-frame embedding space, which is useful for video alignment/correspondence tasks. To the best of our knowledge, CycConf is the first self-supervised task that operates on the instance features and improves the robustness of object detectors.

3. Object Detection with Cycle Confusion

Domain shifts, such as various weather conditions, time of day, viewpoints and geo-locations, are one of the key issues in perception systems. We design robust object detectors that are resistant to domain shifts. Other aspects of model robustness, such as adversarial examples and corruptions, are not covered in the scope of this work.

We improve the out-of-domain generalization of object detectors, which mitigates the performance degeneration when evaluating the object detectors in unseen test domains different from the source domain. That is, an object detector $\mathcal{F}$ is trained on a source domain $\mathcal{D}_s$ (e.g., daytime, highway) and is evaluated on a different test domain $\mathcal{D}_t$ (e.g., night time, city street), where the test domain is *unknown* during training. The robustness of the object detector to domain shifts is measured by the detection average precision (AP) degeneration on the test domain $\mathcal{D}_t$, compared to the model trained on the in-domain data.

This problem setup differs from the typical unsupervised domain adaptation (UDA) setup [1, 3, 17, 21, 33, 51], which considers a projected adaptation between known source and target domains. In UDA, the target domain data is available for training though the labels are missing. In this work, we focus on the out-of-domain generalization setting, and we also show in the experiment sections that our approach can be extended to the UDA setting and outperform specially designed feature alignment based approaches.

3.1. Model Overview

We consider training an object detector in tandem with the auxiliary self-supervised task, CycConf, as shown in Figure 2. CycConf and the object detector share the feature extractor and are jointly trained from scratch. In contrast to pre-training on unlabeled data using self-supervised tasks, we view the self-supervised task as regularization and train the model in a multitask manner.

The CycConf head can be replaced with other self-supervised tasks like rotation and Jigsaw, which share the low-level image features with the object detector rather than instance features as CycConf does.

Base object detector. We adopt the two-stage object detector, Faster R-CNN [32], as the base object detector, which can be replaced with other off-the-shelf detectors. A typical region-based object detector is composed of three components. A backbone feature extractor $\mathcal{B}$ is used to obtain image features. A set of region proposals $\mathbf{p}$ is obtained from a region proposal network (RPN). Through ROI Align [32], we obtain a set of ROI features, which are then fed into the box predictors for box classification and localization.

CycConf uses the region features produced by the RPN network as input while other self-supervised tasks, such as rotation and Jigsaw, use the image features produced by the backbone feature extractor. The objective function to train the entire model is defined as

$$\min \mathcal{L} = \min \left[\mathcal{L}_{det} + \gamma \cdot \mathcal{L}_{sup} \right], \tag{1}$$

where γ is the scaling factor when combining the detection loss $\mathcal{L}_{det}$ and the self-supervised loss $\mathcal{L}_{sup}$. We use $\gamma = 0.01$ in our experiments if not explicitly mentioned.

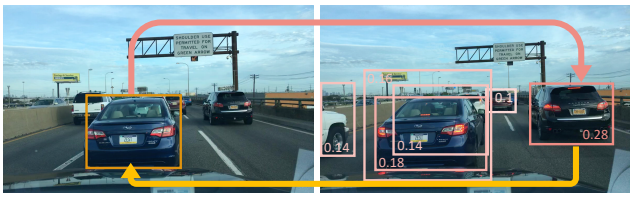
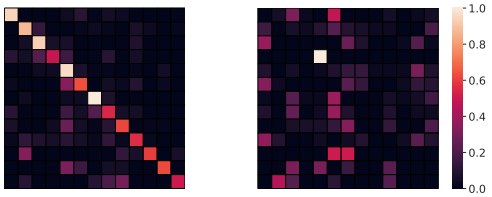


Figure 3: Formation of a time cycle. In the forward pass, the object in the first frame is matched to the most different target in the second frame. The soft target is then matched back to the original object itself in the backward pass to form a cycle.



(a) Nearest Neighbors (b) CycConf

Figure 4: Forward matching probabilities of instance features. Each row represents a different instance. The matching probability distribution of CycConf is more flat and scattered, which means its entropy value is higher than nearest neighbors'.

3.2. Instance-Level Cycle Confusion

In this section, we dive into the details of the instance-level cycle confusion (CycConf) task.

Form a time cycle. CycConf operates on the region features produced by the region proposal network (RPN). For a pair of images at time stamps t_0 and t_1, we collect a set of proposals for each object with objectiveness score above a threshold S. As shown in Figure 3, we consider a matching cycle in time. For an instance I_0 at t_0, in the forward pass from t_0 to t_1, we find a matching target I_1 at t_1. In the backward pass from t_1 to t_0, I_1 should match back to original instance I_0 at t_0 as the self-supervision signal.

Find the soft target. To determine the matching objects across frames, we adopt a similarity measurement using the L_2 distance between two instance features. That is,

$$s_{i,j} = ||\mathbf{u}_i^0 - \mathbf{v}_j^1||_2^2, \tag{2}$$

where $\mathbf{u}_i^0$ is the i-th instance feature at t_0 and $\mathbf{v}_j^1$ is the j-th instance feature at t_1. $\mathbf{u}_i^0$ and $\mathbf{v}_j^1$ are single-dimension vectors obtained by a small encoder network using the region of interest (ROI) features as input. In our experiments, the encoder consists two convolutional layers with kernel size of 3×3 and an average pooling layer. The larger $s_{i,j}$ is, the more different the instances features are.

Following the practice in the cycle consistency literature [7, 44], we consider a *soft matching* target, which is a weighted average of the instance features at t_1 to avoid the instability caused by matching to a single instance feature. The weights α are defined as the normalized exponential of the similarity scores as follows.

$$\hat{\mathbf{v}}^i = \sum_{j=1}^{N_1} \alpha_{i,j} \mathbf{v}_j^1, \quad \alpha_{i,j} = \frac{e^{s_{i,j}}}{\sum_{k=1}^{N_1} e^{s_{i,k}}}, \tag{3}$$

where $\hat{\mathbf{v}}^i$ is the soft target of the i-th instance at t_0 and N_1 is the number of object proposals at t_1.

Since we have a fixed confidence score threshold S to select object proposals, the size of N_1 increases and the matching task becomes more challenging as the object detector is better trained to produce high quality object proposals. Naturally, CycConf forms a curriculum training scheme.

Match back to self. Similarly, the backward pass requires the soft target at t_1 to match to the original object at t_0. We view this backward matching as a classification task, where a cross-entropy loss is adopted as the training objective. We use similarity scores $\mathbf{s}^i \in \mathbb{R}^{N_0}$ defined as

$$\mathbf{s}^i = [s_{1,i}, ...s_{k,i}, ..., s_{N_0,i}], \quad s_{k,i} = ||\mathbf{u}_k^0 - \hat{\mathbf{v}}^i||_2^2. \tag{4}$$

The similarity scores are used as the logits and the loss function is defined as

$$\mathcal{L}_{sup}(\mathbf{u}, \hat{\mathbf{v}}^i) = -\sum_{k}^{N_0} \mathbb{1}(k=i) \log\left(\texttt{softmax}(\mathbf{s}^i)_k\right). \tag{5}$$

Taking all the instance features $\mathbf{u}_i^0$ into account, the overall cycle confusion loss is defined as

$$\mathcal{L}_{sup}(\mathbf{u}, \hat{\mathbf{v}}) = \frac{1}{N_0} \sum_{i=1}^{N_0} \mathcal{L}_{sup}(\mathbf{u}, \hat{\mathbf{v}}^i). \tag{6}$$

We train the object detector and the CycConf task jointly by combining the detection loss and the self-supervised task loss (Equation 1).

Maximum entropy interpretation. The matching procedure in CycConf can be interpreted as a maximum entropy regularization. In Figure 4, we visualize the matching probability distribution in the forward pass of CycConf and compare it with nearest neighbors on the instance features. We can see from the figure, nearest neighbors tends to match the identical objects across the frames since the diagonal instances have much higher matching probability. In contrast, the matching probability distribution in CycConf is flat and scattered, whose entropy value is higher than nearest neighbors'. In technical terms, CycConf can be viewed as maximizing the entropy of the probability distribution of the weights α,

$$\max H(\alpha) = -\frac{1}{N_0} \sum_{i=0}^{N_0} \sum_{j=1}^{N_1} \alpha_{i,j} \log(\alpha_{i,j}). \tag{7}$$

(a) BDD100K Daytime (b) BDD100K Night (c) Waymo Front Left (d) Waymo Side Left

Figure 5: Training samples from BDD100K and Waymo Open datasets. The Waymo open dataset has different camera angles as BDD100K.

Table 1: Dataset statistics of BDD100K and Waymo Open data.

Dataset	Split	seq.	frames/seq.	boxes	classes
BDD100K Daytime	Train	757	204	1.82M	8
	Val	108	204	287K	8
BDD100K Night	Train	564	204	895K	8
	Val	71	204	137K	8
Waymo Open Data	Train	798	199	3.64M	3
	Val	202	199	886K	3

The principle of maximum entropy [18, 19] indicates the probability distribution with the highest entropy is the one that best represents the current state of knowledge in the context of precisely stated prior data, which in the recent literature [8, 30, 41, 52] shows can improve model robustness.

4. Out-of-Domain Evaluation

In this section, we describe the out-of-domain evaluation benchmark for object detectors using large-scale driving video datasets BDD100K [48] and the Waymo Open Dataset [39] in Section 4.1. We show that CycConf outperforms other commonly used self-supervised tasks when evaluated on the unseen test domains in Section 4.2. Visualization and analysis are presented in Section 4.3.

4.1. Benchmark Construction

Datasets. For BDD100K, we divide the dataset into non-overlapping daytime and night splits to create a domain gap. We denote the splits as *BDD100K Daytime* and *BDD100K Night*. For the Waymo Open data, we split the dataset based on five different angles including front, front left, front right, side left, and side right. The front camera angle is consistent with the camera angle in the BDD100K dataset, while the other camera angles are not. The statistics of all the datasets are provided in Table 1 and some example training samples are provided in Figure 5.

Evaluation settings. We consider two out-of-domain evaluation scenarios: (1) *Domain Shift by Time of Day*, where the model is trained and evaluated on different time of day. (2) *Cross-camera domain shift*, where the model is trained and evaluated under different camera views. Since the Waymo Open data and BDD100K are collected from different sensors, the dataset distribution shift is more severe than the first scenario.

Baselines. We consider three self-supervised tasks and compare them with the proposed Faster R-CNN w/ CycConf.

- Faster R-CNN w/ Rotation [10]. We jointly train Faster R-CNN with the image-level rotation task, where each image is rotated, and the detector additionally has to predict the angle of rotation.
- Faster R-CNN w/ Jigsaw [26]. For Jigsaw, a 2x2 grid is sampled from each image and shuffled, and the detector has to predict the permutation of the tiles.
- Faster R-CNN w/ Cycle Consistency. The instance-level cycle consistency task is to find the nearest neighbors across the consecutive frames. This is a baseline method we adapted from the existing literature [7], where we replace the random patches with the instance features produced by the object detector.

Implementation details. We conduct the experiments using PyTorch 1.6.0 [28] with Detectron2 [45] library. For all experiments, we use Faster R-CNN [32] as our base detector and Resnet-50 [11] with a Feature Pyramid Network [24] as the backbone. All models are trained on 8 GPUs using SGD with a mini-batch size of 16, momentum of 0.9, and weight decay of 0.0001. We set the object score threshold $S = 0.8$ in our experiments if not mentioned.

4.2. Evaluation Results

Domain Shift by Time of Day. We first evaluate on the large scale BDD100K dataset. We construct two settings on this dataset, BDD100K Daytime $\rightarrow$ Night and BDD100K Night $\rightarrow$ Daytime. For each setting, we train our detector on one split and evaluate on the other split. The results for BDD100K are shown in Table 2.

In both settings, our proposed CycConf method is able to significantly improve the base detector's performance. In particular, on BDD100K Daytime $\rightarrow$ Night, CycConf can achieve around 2 points improvement in AP50 and 1.5 points improvement in AP over the base detector. Compared to other self-supervised tasks, CycConf can also lead to better performance across both settings. CycConf is able to outperform the cycle consistency baseline consistently, indicating that CycConf helps the detector generalize better compared to cycle consistency.

Table 2: BDD100K Daytime and Night. CycConf outperforms other self-supervised tasks in both settings.

	BDD100K Daytime → Night						BDD100K Night → Daytime					
Model	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
Faster R-CNN	17.84	31.35	17.68	4.92	16.15	35.56	19.14	33.04	19.16	5.38	21.42	**40.34**
+ Rot	18.58	32.95	18.15	5.16	16.93	**36.00**	19.07	33.25	18.83	5.53	21.32	40.06
+ Jigsaw	17.47	31.22	16.81	5.08	15.80	33.84	19.22	33.87	18.71	5.67	22.35	38.57
+ Cycle Consistency	18.35	32.44	18.07	5.04	17.07	34.85	18.89	33.50	18.31	5.82	21.01	39.13
+ Cycle Confusion (Ours)	**19.09**	**33.58**	**19.14**	**5.70**	**17.68**	35.86	**19.57**	**34.34**	**19.26**	**6.06**	**22.55**	38.95

Table 3: Waymo to BDD100K. The domain gap is due to the changing camera angles. CycConf outperforms other self-supervised tasks in both settings by a relatively large margin.

	Waymo Front Left → BDD100K Night						Waymo Front Right → BDD100K Night					
Model	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
Faster R-CNN	10.07	19.62	9.05	2.67	10.81	18.62	8.65	17.26	7.49	1.76	8.29	19.99
+ Rot	11.34	23.12	9.65	**3.53**	11.73	21.60	9.25	18.48	8.08	1.85	8.71	**21.08**
+ Jigsaw	9.86	19.93	8.40	2.77	10.53	18.82	8.34	16.58	7.26	1.61	8.01	18.09
+ Cycle Consistency	11.55	23.44	10.00	2.96	12.19	21.99	9.11	17.92	7.98	1.78	9.36	19.18
+ Cycle Confusion (Ours)	**12.27**	**26.01**	**10.24**	3.44	**12.22**	**23.56**	**9.99**	**20.58**	**8.30**	**2.18**	**10.25**	20.54

Cross-camera Domain Shift. We construct two settings on this dataset, Waymo Front Left → BDD100K Night and Waymo Front Right → BDD100K Night. Waymo Front Left and Waymo Front Right consist of images taken from the front left and the front right cameras, respectively. We train our detector on the Waymo data and then evaluate on BDD100K Night. Since there are changes in the view angle, the domain gap in these two settings is larger than that in the previous settings.

The results for Waymo to BDD100K are shown in Table 3. Despite the larger domain gap, our proposed CycConf is still able to outperform the other methods by a relatively large margin. On Waymo Front Right → BDD100K Night, CycConf can achieve around 2 points improvement in AP50 and 1 point in AP. In both settings, CycConf is able to significantly improve the performance of the base detector by over 3 points in AP50. CycConf also performs consistently better than the model trained with cycle consistency.

4.3. Visualization and Analysis

Proposal matching. We visualize proposal matches in Figure 6 to show the difference between the behavior of instance-level cycle consistency and CycConf. For the former, each proposal is encouraged to be matched to its nearest neighbor in the next frame, and thus it tends to be matched to itself in the next frame. For CycConf, each proposal is matched to the most different proposal in the next frame. In the first sequence (top row), the car on the left is matched to the car on the right. In the second sequence (bottom row), the car in the front is matched to the green car on the left.

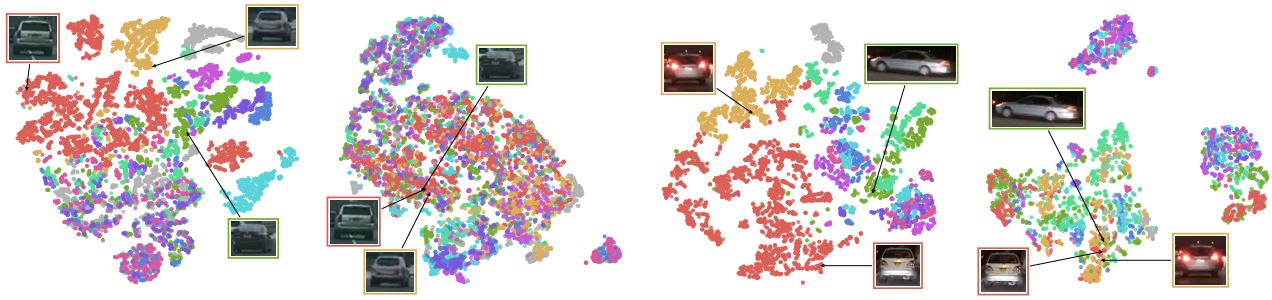
Instance features. We use t-SNE to visualize the instance features using cycle consistency and CycConf on BDD100K Daytime → Night in Figure 7. We extract features of proposals that are matched to ten car instances in a sequence of images from both datasets. Each color represents features of a different instance. We additionally show example images of three instances and the position of their features in the t-SNE space. On both datasets, the instance features learned with cycle consistency are well separated despite the visual similarity between the different instances. On the other hand, CycConf maps similar objects to features that are more mixed together. This makes it more difficult for the detector to identify the instance identities and encourages the detector to explore the latent structures of each instance.

(a) Cycle Consistency (b) Cycle Confusion

Figure 6: Examples of proposal matches of the detector trained with cycle consistency and CycConf on BDD100K. When using cycle consistency, the matched proposals are similar, while CycConf matches dissimilar proposals.

5. Unsupervised Domain Adaptation

In this section, we extend the study to typical unsupervised domain adaptation (UDA) benchmarks, a closely related setting, to test the out-of-domain generalization. To the best of our knowledge, existing UDA methods



(a) Cycle Consistency (Daytime) (b) CycConf (Daytime) (c) Cycle Consistency (Night) (d) CycConf (Night)

Figure 7: t-SNE visualization of instance features from Cycle Consistency ((a) and (c)) and CycConf ((b) and (d)) on BDD100K Daytime $\rightarrow$ Night. Each color represents a different instance. Images of three instances are shown for each dataset as well as the position of their features in the t-SNE space. The features of different instances when using the cycle consistency task are well separated despite their visual similarity. In comparison, CycConf's features are more mixed, which encourages the detector to explore the latent structures of each instance.

rely on the availability of a handful of images from the target domain, and the domain adaptive object detectors [1, 3, 17, 21, 33, 51] are mostly evaluated on static images and not videos. To conduct a fair comparison, we train the object detector jointly with the self-supervised tasks on existing UDA benchmarks with static images. The self-supervised task used in this section is image rotation, as CycConf is designed to operate on videos. The key message in these experiments is that the adoption of the self-supervised tasks can be a general solution to multiple evaluation benchmarks, whether out-of-domain generalization or typical UDA benchmarks. As shown in Section 5.1, a joint training approach has better accuracy than the previous feature alignment based approaches with much less training cost. We hope this finding can motivate future algorithm designs.

Weather adaptation. We first adapt the model from Cityscapes [4] to Foggy Cityscapes [34]. The Cityscapes dataset is a driving scene dataset, containing 2,975 and 500 images in the training and validation set, respectively, while the Foggy Cityscapes data is created by adding synthetic fog to Cityscapes. There are eight classes: *person, rider, car, truck, bus, train, motorcycle and bicycle* in both dataset.

Synthetic-to-real. Sim10K [20] contains 10K synthesized images with bounding box annotations. We use images of Sim10K as the source domain and adapt the model to the Cityscapes dataset. Following previous works [3, 17, 51], only the *car* class is considered.

Implementation details. The base object detector is Faster RCNN [32] with FPN [25] and uses VGG-16 [37] and ResNet-101 [11] as the backbones.

We use rotation [10] as the self-supervised task and perform image-level joint training with the detector. We randomly crop the input image to a 224$\times$224 patch and select a rotation angle from 0°, 90°, 180° and 270° to construct the input and label for the self-supervised task. We jointly optimize the model with the detection loss and the self-supervised loss on the source domain data and only with the self-supervised loss on the target domain data. We set the loss scale λ of the rotation task to 0.5.

We use 8 GPUs for training, a batch size of 32, and the base learning rate of 0.01. We train the model for 10K iterations in total and divide the learning rate by 10 at 6K and 8K iterations. For evaluation metrics, we report the mean average precision (mAP) with an intersection of union (IoU) threshold of 0.5.

5.1. Evaluation Results

Weather adaptation. We evaluate Faster R-CNN w/ rotation on the Cityscapes$\rightarrow$ Foggy Cityscapes benchmark and show the results in Table 4. Our model consistently outperforms pervious approaches with both VGG-16 and ResNet-101 as backbones. We can achieve an mAP of 41.5 points, improving the base Faster R-CNN model by 15.9 points. It also outperforms the prior art, EPM, by 1.3 points. Similar results apply to using VGG-16 as backbone. Faster R-CNN improves the base Faster R-CNN from 18.8 to 37.8, an absolute improvement of 19 points in mAP. It also outperforms the prior art by $\sim$2 points. These results indicate that Faster R-CNN w/ Rot can effectively handle the domain shift despite its architectural simplicity.

We also report the average precision of each category. We consistently improve the prior art EPM for all classes except *truck* and *train*, which have limited annotations available (466 and 158 annotations). More complicated adaptation strategies might be useful for these rare classes and we leave detailed study for future work.

Synthetic-to-real. We additionally compare our method with previous approaches under the synthetic-to-real setting shown in Table 5. The model is trained on the Sim10K dataset with full annotation together with the unlabeled data from the training set of Cityscapes. We evaluate on the validation set of Cityscapes and report the mAP50 for *car*,

Table 4: Adapt from Cityscapes to Foggy Cityscapes. We report the AP50 of 8 categories and the mean AP across all classes. We adopt both VGG-16 and ResNet-101 as the backbone. Despite its simplicity, Faster R-CNN w/ Rot outperforms the previous methods by a large margin.

Cityscapes→Foggy Cityscapes										
Method	Backbone	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP50
Faster R-CNN (source only) [3]	VGG-16	17.8	23.6	27.1	11.9	23.8	9.1	14.4	22.8	18.8
DAF [3]		25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6 (+8.8)
MAF [12]		28.2	39.5	43.9	23.8	39.9	**33.3**	29.2	33.9	34.0 (+15.2)
SW-DA [33]		29.9	42.3	43.5	24.5	36.2	32.6	**30.0**	35.3	34.3 (+15.5)
DAM [21]		30.8	40.5	44.3	**27.2**	38.4	34.5	28.4	32.2	34.6 (+15.8)
EPM [17]		41.9	38.7	56.7	22.6	41.5	26.8	24.6	35.5	36.0 (+17.2)
Faster R-CNN (oracle)		47.4	40.8	66.8	27.2	48.2	32.4	31.2	38.3	41.5 (+22.7)
Faster R-CNN w/ Rot	VGG-16	**42.2**	**47.2**	**59.8**	23.2	**43.5**	19.8	27.2	**40.0**	**37.8 (+19.0)**
Faster R-CNN (source only) [17]	ResNet-101	33.8	34.8	39.6	18.6	27.9	6.3	18.2	25.5	25.6
EPM [17]		41.5	43.6	57.1	**29.4**	44.9	**39.7**	29.0	36.1	40.2 (+14.6)
Faster R-CNN (oracle)		52.3	55.8	73.8	37.8	54.4	31.3	36.4	47.3	48.6 (+23.0)
Faster R-CNN w/ Rot	ResNet-101	**45.8**	**51.0**	**63.1**	26.8	**47.1**	23.6	**30.6**	**43.6**	**41.5 (+15.9)**

Table 5: Adapt from Sim10K to Cityscapes (S→C). Faster R-CNN w/ Rot outperforms the previous approaches consistently.

Method	Backbone	S→C (mAP50)
Faster R-CNN (source only)	VGG-16	30.1
DAF [3]		39.0 (+8.9)
MAF [12]		41.1 (+11.0)
SW-DA [33]		42.3 (+12.2)
SW-DA* [33]		47.7 (+17.6)
SC-DA [51]		43.0 (+12.9)
EPM [17]		49.0 (+18.9)
Faster R-CNN (oracle)		69.7 (+39.6)
Faster R-CNN w/ Rot	VGG-16	**50.1 (+20.0)**
Faster R-CNN (source only)	ResNet-101	41.8
EPM [17]		51.2 (+9.6)
Faster R-CNN (oracle)		70.4 (+28.6)
Faster R-CNN w/ Rot	ResNet-101	**52.4 (+10.4)**

as this is the only class in Sim10k dataset.

Our model outperforms all the other methods with both VGG-16 and ResNet-101 as backbones. With ResNet-101 as our backbone, we improve the prior art EPM from 51.2 to 52.4. With VGG-16 as backbone, our model can also improve upon EPM by 1.1 points.

Training time comparison. We provide an estimation of the training time of various approaches in Figure 8 by running the released source code of previous approaches in the same environment. We find that our method reduces the training time of EPM by half while achieving higher AP scores, which is largely due to the simplicity of our approach.

6. Conclusion

In this work, we investigated the usage of auxiliary self-supervised tasks to improve the robustness of object detectors under domain shifts. Through extensive study on various benchmarks, we show that auxiliary self-supervised tasks, used in tandem with the object detection training, are effective to improve the out-of-domain generalization of structural prediction tasks and that they can be used across different settings, whether abundant test domain data are available. The results can be inspiring as a general direction to improve the model robustness under distribution shifts. We also introduced instance-level cycle confusion (CycConf), a self-supervised task on the region features produced by the object detector. For each object, the task is to find the most different object proposals in the adjacent frame in a video and then cycle back to itself for self-supervision. CycConf encourages the object detector to explore invariant structures across instances under various motion, viewpoint, lightning, etc., which leads to improved model robustness in unseen domains at test time. Our model establishes a new state-of-the-art on large scale video benchmarks.

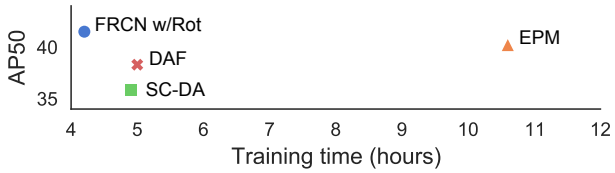


Figure 8: Training time and AP of various approaches. We train the previous approaches with ResNet-101 as the backbone on 8 GPUs for weather adaptation. Faster R-CNN w/ Rot reduces the training time of EPM by half while achieving higher AP scores.

Acknowledgments

This work was supported by RISE Lab, Berkeley AI Research, Berkeley DeepDrive and DARPA. This work was supported in part by DoD including DARPA's XAI, LwLL, and/or SemaFor programs, as well as BAIR's industrial alliance programs. In addition to NSF CISE Expeditions Award CCF-1730628, this research was supported by gifts from Alibaba, Amazon Web Services, Ant Financial, CapitalOne, Ericsson, Facebook, Futurewei, Google, Intel, Microsoft, Nvidia, Scotiabank, Splunk and VMware. Prof. Wang's group was supported, in part, by gifts from Qualcomm and TuSimple.

References

[1] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019. 1, 2, 3, 7

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 1

[3] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 1, 2, 3, 7, 8

[4] 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

[5] Murat Dikmen and Catherine M Burns. Autonomous driving in the real world: Experiences with tesla autopilot and summon. In *Proceedings of the 8th international conference on automotive user interfaces and interactive vehicular applications*, pages 225–228, 2016. 1, 2

[6] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019. 2, 4, 5

[8] Benjamin Eysenbach and Sergey Levine. Maximum entropy rl (provably) solves some robust rl problems, 2021. 2, 5

[9] 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 *International Conference on Learning Representations*, 2019. 2

[10] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR 2018*, 2018. 1, 2, 5, 7

[11] 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. 5, 7

[12] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6668–6677, 2019. 8

[13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018. 1, 2

[14] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019. 1, 2

[15] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 2

[16] 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. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1, 2

[17] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *European Conference on Computer Vision*, pages 733–748. Springer, 2020. 1, 2, 3, 7, 8

[18] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957. 2, 5

[19] Edwin T Jaynes. Information theory and statistical mechanics. ii. *Physical review*, 108(2):171, 1957. 2, 5

[20] 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? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 746–753. IEEE, 2017. 7

[21] 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 Conference on Computer Vision and Pattern Recognition*, pages 12456–12465, 2019. 1, 2, 3, 7, 8

[22] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton A. Earnshaw, Imran S. Haque, Sara Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. WILDS: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning (ICML)*, 2021. 1, 2

[23] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017. 1, 2

[24] Tsung-Yi Lin, Piotr Dollar, 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. 5

[25] 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. 7

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 1, 2, 5

[27] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[29] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[30] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems VIII*, 2012. 2, 5

[31] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018. 2

[32] 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. 1, 3, 5, 7

[33] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019. 1, 2, 3, 7, 8

[34] 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. 7

[35] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1134–1141. IEEE, 2018. 2

[36] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. A systematic framework for natural perturbations from videos. 2019. 2

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Machine Learning*, 2015. 7

[38] Peng Su, Kun Wang, Xingyu Zeng, Shixiang Tang, Dapeng Chen, Di Qiu, and Xiaogang Wang. Adapting object detectors with conditional domain normalization. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[39] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2, 5

[40] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019. 2

[41] Emanuel Todorov et al. Linearly-solvable markov decision problems. In *NIPS*, pages 1369–1376, 2006. 2, 5

[42] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008. 2

[43] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 2

[44] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 2, 3, 4

[45] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5

[46] 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. 2

[47] 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. 2

[48] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 5

[49] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[50] Ganlong Zhao, Guanbin Li, Ruijia Xu, and Liang Lin. Collaborative training between region proposal localization and classi? cation for domain adaptive object detection. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[51] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019. 1, 2, 3, 7, 8

[52] Brian D Ziebart. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010. 2, 5