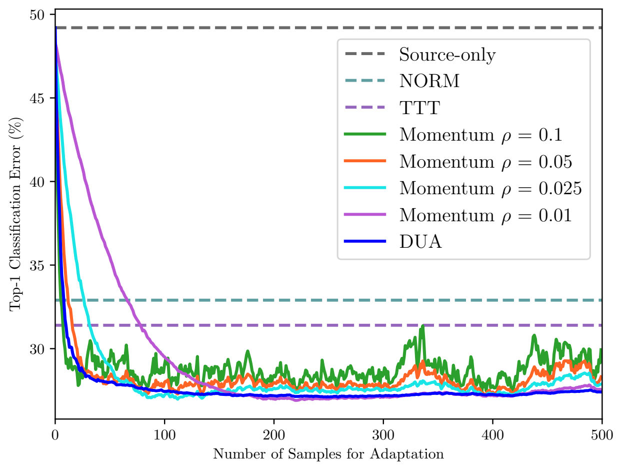
The Norm Must Go On: Dynamic Unsupervised Domain Adaptation by Normalization

M. Jehanzeb Mirza[1,2] Jakub Micorek[1] Horst Possegger[1] Horst Bischof[1,2]

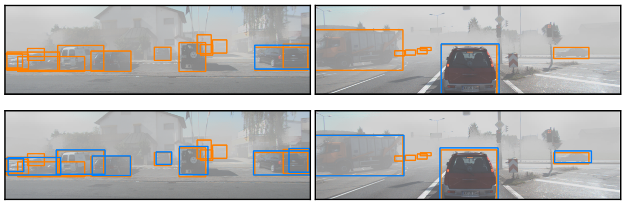
[1]Institute of Computer Graphics and Vision, Graz University of Technology.
[2]Christian Doppler Laboratory for Embedded Machine Learning.
{muhammad.mirza, jakub.micorek, possegger, bischof}@icg.tugraz.at

Abstract

Domain adaptation is crucial to adapt a learned model to new scenarios, such as domain shifts or changing data distributions. Current approaches usually require a large amount of labeled or unlabeled data from the shifted domain. This can be a hurdle in fields which require continuous dynamic adaptation or suffer from scarcity of data, e.g. *autonomous driving in challenging weather conditions. To address this problem of continuous adaptation to distribution shifts, we propose Dynamic Unsupervised Adaptation (DUA). By continuously adapting the statistics of the batch normalization layers we modify the feature representations of the model. We show that by sequentially adapting a model with only a fraction of unlabeled data, a strong performance gain can be achieved. With even less than 1% of* unlabeled data *from the target domain, DUA already achieves competitive results to strong baselines. In addition, the computational overhead is minimal in contrast to previous approaches. Our approach is simple, yet effective and can be applied to any architecture which uses batch normalization as one of its components. We show the utility of DUA by evaluating it on a variety of domain adaptation datasets and tasks including object recognition, digit recognition and object detection.*



(a) CIFAR-10C results.



(b) Detections in foggy weather without (top) and with (bottom) DUA. We overlay the detection results (blue) and the ground truth (orange).

Figure 1. Exemplary DUA results. a) Mean classification error over 15 different corruption types (at the most severe level 5) on CIFAR-10C [18]. We outperform the state-of-the-art NORM [42, 54] and TTT [60], while using less than 1% of unlabeled data from the corrupted test set only. Our proposed adaptive momentum scheme leads to both fast and stable improvements in contrast to fixing the momentum parameter ρ. b) Qualitative results for object detection in degrading weather conditions: our DUA (bottom) significantly improves the performance of a KITTI [11] pre-trained YOLOv3 [48] on KITTI-Fog [14]. Best viewed in color.

1. Introduction

Present day Deep Neural Networks (DNNs) show promising results when both training and testing data belong to the same distribution [16, 26, 68]. However, if there is a domain shift, *i.e.* when the testing data comes from a different domain, neural networks struggle to generalize [4, 10, 38]. In fact, even if there is only a slight distribution shift, the performance of neural networks is reported to already degrade significantly [18, 47].

One way to overcome the performance drop during domain shifts is to obtain labeled data from the shifted domain and re-train the network. However, manual labeling of large amounts of data imposes significant human and monetary costs. These issues are addressed by Unsupervised Domain Adaptation (UDA) approaches, *e.g.* [4, 10, 12, 17, 24, 33, 34, 52, 67, 75]. For UDA, the goal is to modify the network parameters in such a way that it can adapt in an unsupervised manner to out-of-distribution testing data. Traditionally, these approaches require labeled training data along with a large amount of unlabeled testing data.

In many practical scenarios, the traditional requirements, *i.e*. access to both labeled training and large amounts of unlabeled testing data can often not be fulfilled. For example, in the medical domain, pre-trained models are often provided without access to the training data (which is kept private due to privacy regulations). Likewise, some application domains benefit from dynamic adaptation to a changing environment. For example, consider object detectors for autonomous vehicles, which are usually trained on mostly clear weather images, *e.g*. [11, 15, 58]. In real-world scenarios, however, weather can suddenly deteriorate, resulting in significant performance degradation [40, 41]. In such cases, it is not feasible to obtain labeled training data captured in degrading weather and re-train the detector from scratch. A better solution is to dynamically adapt the detector, given only a few (unlabeled) bad weather examples.

In this work, we highlight that one hindrance in domain generalization is the statistical difference in mean and variance between train and (shifted) test data. Thus, during inference, we adapt the running mean and variance which are calculated during training by the batch normalization layer [21]. Moreover, we adapt the statistics dynamically in an online manner on a tiny fraction of test data. For adaptation, we form a small batch by augmenting each incoming sample. In order to ensure stable adaptation and fast convergence we propose an adaptive update schema.

Related approaches [28, 42, 54, 63] typically ignore the training statistics and recalculate the batch statistics from scratch for the test data. This, however, requires large batches of test data. We argue that a large batch of test data might not always be available in real-world applications, *e.g*. autonomous cars adapting to challenging weather (see Figure 1). We show that a strong performance gain can be achieved by adapting the running mean and variance in an online manner (one sample at a time). In particular, we require only a small number of sequential samples from the out-of-distribution data.

Our contributions can be summarized as follows:

- We show that online adaptation of batch normalization parameters on a tiny fraction of unlabeled out-of-distribution test data can provide a strong performance gain. With even less than 1% of unlabeled test data, DUA already performs competitively to strong baselines which use the entire test set for adaptation.
- DUA is simple, unsupervised, dynamic and requires no back propagation [50] to work. Since the computational overhead is also negligible, it is perfectly suited for real-time applications.
- We evaluate DUA on a variety of domain shift benchmarks, demonstrating its beneficial performance. We achieve state-of-the-art results on most benchmarks while being competitive on the remaining.
- We show that our dynamic adaptation method works on a variety of different tasks and different architectures. To the best of our knowledge, we are the first to show dynamic adaptation for object detection.

2. Related Work

Unsupervised Domain Adaptation (UDA) has received a significant amount of interest recently. We summarize these approaches in four categories: minimizing discrepancy between domains, adversarial approaches, self-supervised approaches and correcting domain statistics.

Discrepancy reduction between source and target domains is usually performed at specific network layers or in a contrastive manner. Long et al. [37] match the mean embeddings from task specific layers. Sun et al. [56, 57] minimize the second order statistics to align the source and target domains. They apply a linear transformation on the source domain to align it with the target domain. Zellinger et al. [73] propose to match higher order moments by introducing a Central Moment Discrepancy (CMD) metric to learn domain invariant features. Chen et al. [2] propose to match third and fourth order statistics of the source and target domains for unsupervised domain adaptation. On the other hand, [23, 64] use contrastive learning [6] to reduce discrepancy between domains.

Adversarial discriminative approaches align the features from the source and target domains mostly by using the domain confusion loss. Ganin et al. [10] propose a method which is based on the philosophy that predictions must be made on features which are non-discriminative during training. A novel gradient reversal layer is proposed which brings the features from the source and target domains closer by maximizing the domain confusion loss. Tzeng et al. [62] also rely on maximizing the domain confusion loss for unsupervised domain adaptation. Hong et al. [19] use a fully convolutional network and use generative adversarial networks [13] to address the problem of synthetic-to-real feature alignment. Chen et al. [5] align the global and class wise features by using a generative adversarial network. Similar approaches for UDA have also been followed for a variety of tasks, including object detection [4, 8, 17, 24, 65, 67, 69, 70, 75], object classification [30, 33, 34, 36, 46] and semantic segmentation [1, 3, 22, 29, 71, 76] for both 2D and 3D data.

Self Supervision has also been used for the purpose of unsupervised domain adaptation. Sun et al. [59] combine different self supervised auxiliary tasks for domain adaptation. Sun et al. [60] also propose Test Time Training (TTT) with self supervision. They put forward the idea of removing the self-imposed condition of a fixed decision boundary at test time. In their work they use the rotation prediction task [12] as a self supervised task in order to adapt the network to out-of-distribution test data.

Correcting the domain statistics calculated by the batch normalization layer [21] has also been used for UDA. Li et al. [28] propose Adaptive Batch Normalization where they show that recalculating the batch normalization parameters from scratch for the test set can improve generalization of DNNs. Carlucci et al. [39] propose domain adaptation layers which can learn a hyperparameter during training to find the optimal mixing of statistics from the source and target domain. Singh et al. [55] study the effect of lower batch sizes during training and show that DNNs using batch normalization layers are effected by lower batch sizes. They propose an auxiliary loss for remedy. Similarly, [42, 54, 74] also show that recalculation of batch normalization statistics from scratch for test data can be helpful to address the problem of distribution shift between source and target domains. Wang et al. [63] also recalculate the batch normalization statistics for the test data. Further, they calculate the loss from the entropy of predictions and adapt the scale and shift parameters of batch normalization layers. It is important to point out that [28, 42, 54, 63] share the same philosophy of a variable decision boundary at test time as TTT [60].

Our work closely resonates with [28, 42, 54, 63] and is also similar in philosophy to TTT [60], aiming for a variable decision boundary at test time. However, we differ from them in several fundamental ways: In [28, 42, 54, 63], the training statistics are ignored and the batch statistics are recalculated from the test set. For this reason they require large batches from the test set. However, in general, large batches of test data might not be available. Instead, we adapt the statistics calculated from the training data in an online manner (on each incoming sample). We show competitive results by using less than 1% of unlabeled test data in contrast to all previous approaches which use the complete test set. Further, contrary to previous approaches, such as [60, 63], our method does not require back propagation. Our scenario is more realistic for dynamic adaptation where we can obtain only a single test frame at one time.

3. Approach

We first summarize batch normalization [21] in Section 3.1 as it lies at the center of our approach. Section 3.2 then details our DUA approach.

3.1. Batch Normalization

Ioffe and Szegedy [21] proposed a batch normalization layer which has become an important component of modern day DNNs. Each batch normalization layer in the network calculates the mean and variance for each activation coming from the training data X, and normalizes each incoming sample x as

$$\hat{x} = \frac{x - \mathbb{E}[X]}{\sqrt{\mathrm{Var}[X] + \epsilon}} \cdot \gamma + \beta, \tag{1}$$

where γ and β are the scale and shift parameters, and ϵ is used for numerical stability. The expected value $\mathbb{E}[X]$ of the training statistics is estimated through the running mean,

$$\hat{\mu}_k = (1 - \rho) \cdot \hat{\mu}_{k-1} + \rho \cdot \mu_k, \tag{2}$$

and the variance of the training statistics $\mathrm{Var}[X]$ is estimated through running variance,

$$\hat{\sigma}_k^2 = (1 - \rho) \cdot \hat{\sigma}_{k-1}^2 + \rho \cdot \sigma_k^2. \tag{3}$$

Here, $\hat{\mu}$ and $\hat{\sigma}^2$ are the estimated mean and variance from the training data, whereas μ and σ^2 represent the mean and variance of the incoming batch. The hyperparameter ρ is the momentum term (default $\rho = 0.1$) and k denotes each training step. Intuitively, ρ can be thought of as the factor which controls how much the existing estimate of statistics is affected by the statistics of the incoming batch. A larger momentum value would essentially give more weight to the calculated statistics of the incoming batch. Empirically, it has been shown that batch normalization helps to train faster and also stabilize the training process [53].

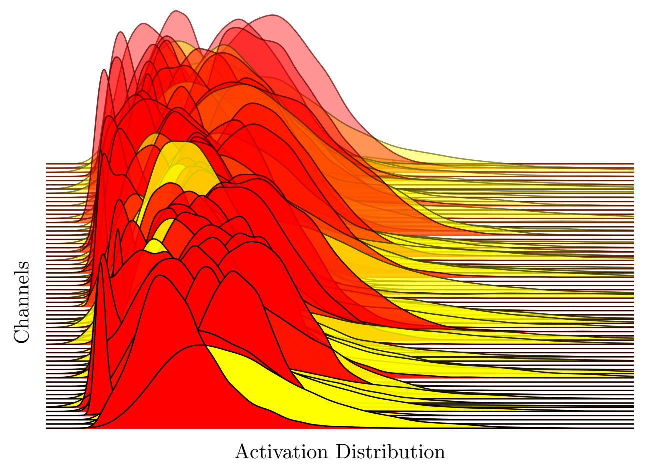
The behavior of batch normalization differs during training and testing as follows:

Training: During training, the batch normalization layer calculates the running mean and variance over the complete training set. The scale parameter γ and shift parameter β from Eq. (1) are learned by back propagation. Running mean and variance is updated during each forward pass with the new batch statistics.

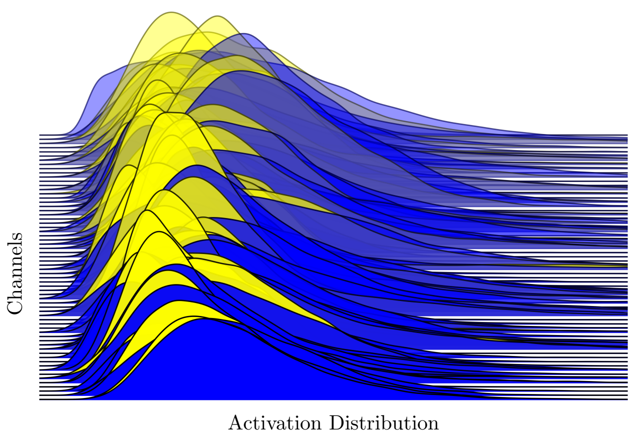
Testing: During inference, the running mean and variance of the batch normalization layer is fixed. Each new sample encountered during testing is normalized by using the population statistics calculated during training.

3.2. Dynamic Unsupervised Adaptation

Let Φ_{src} be the network trained solely with source data X_{src}. Our goal is to adapt the trained model to out-of-distribution target data X_{tar} in an unsupervised manner. The batch normalization layer performs consistently well when train and test data belong to a similar distribution [16, 68]. However, in many practical scenarios this is not the case. It has been shown that when out-of-distribution test data is encountered, batch normalization can hamper the performance significantly [9, 28, 42, 54, 63, 66]. One reason for the



(a) Source Model



(b) Model Adapted with DUA

Figure 2. Density plots of output distribution for the 64 channels of the last batch normalization layer of ResNet-26 trained on CIFAR-10. a) Yellow is the output distribution of the training data. Red is the output distribution of the shifted test data, *i.e.* corrupted with Contrast Level-5 [18]. The misalignment of the feature responses is one reason for the performance drop. b) Yellow is the output distribution of the training data. Blue is the output distribution of the shifted test data, *i.e.* corrupted with Contrast Level-5, after adaptation with DUA. DUA aligns the output distribution from corrupted data closely with the clean (training) distribution. Best viewed in color.

performance degradation is the misalignment of the activation distribution between training and out-of-distribution test data as shown in Figure 2a. Thus, our adaptation process aligns the activation distribution between training and shifted test data as depicted in Figure 2b.

In our proposed adaptation schema, all the parameters of the network Φ_{src}, other than the running mean and running variance are fixed. We only adapt the $\mathbb{E}[X]$ and $\text{Var}[X]$ from Eq. (1) to the new statistics of X_{tar}, by using the training statistics obtained from X_{src} as a prior. The training statistics are updated by using one image after the other, *i.e.* processing new examples of the (shifted) test data in a sequential manner as they arrive. The naïve approach would be to update the statistics by using Eqs. (2) and (3) with a fixed momentum parameter ρ. However, as shown in Figure 1a, such fixed momentum leads to a couple of problems: The adaptation performance is either unstable or converges rather slowly. This is because with the default parameters the adaptation of the running mean and variance is highly unstable, as shown in Figure 3a. Thus, for stable and fast convergence we adapt the momentum with each incoming sample. More formally, we update the mean and variance consecutively:

$$\hat{\mu}_k = (1 - (\rho_k + \zeta)) \cdot \hat{\mu}_{k-1} + (\rho_k + \zeta) \cdot \mu_k, \tag{4}$$

with

$$\hat{\mu}_0 = \hat{\mu}_s, \quad \rho_k = \rho_{k-1} \cdot \omega, \quad \rho_0 = 0.1, \tag{5}$$

and

$$\hat{\sigma}_k^2 = (1 - (\rho_k + \zeta)) \cdot \hat{\sigma}_{k-1}^2 + (\rho_k + \zeta) \cdot \sigma_k^2, \tag{6}$$

with

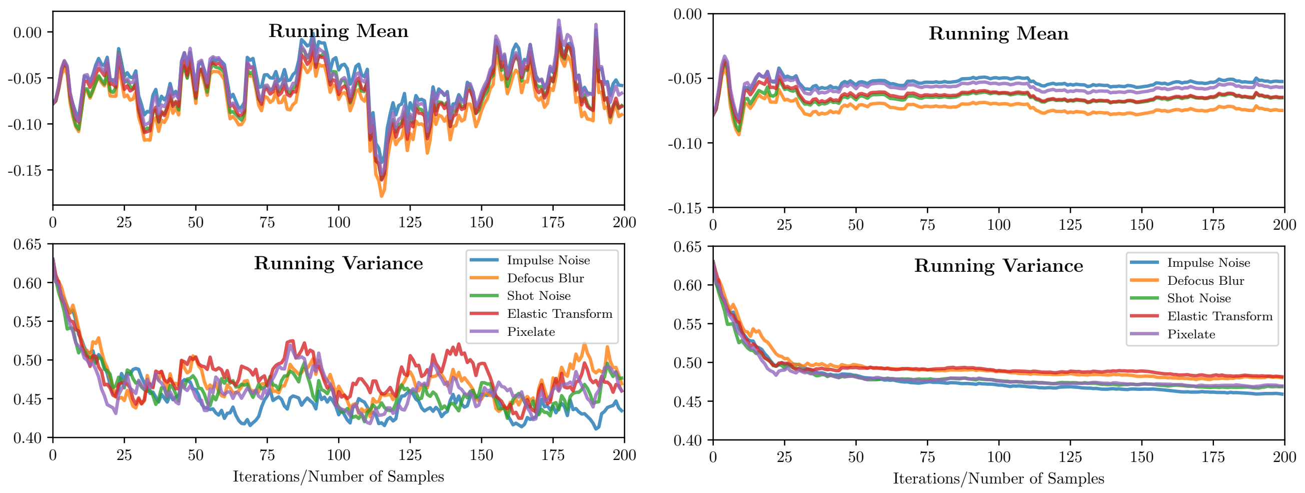
$$\hat{\sigma}_0^2 = \hat{\sigma}_s^2, \quad \rho_k = \rho_{k-1} \cdot \omega, \quad \rho_0 = 0.1. \tag{7}$$

Here, $\omega \in (0, 1)$, is the momentum decay parameter, whereas ζ, with $0 < \zeta < \rho_0$, is a constant and defines the lower bound of the momentum. As the momentum ρ_k decays, the later samples will have a smaller impact. Our adaptive momentum scheme has a direct impact on how the running mean and variance are adapted. The adaptation becomes stable as compared to default parameters, which is visible in Figure 3b.

Whenever we obtain a new sample from the (shifted) test distribution, we make a small batch by augmenting the incoming sample. In particular, we use random horizontal flipping, random cropping and rotation. We take care that the augmentations we use do not correlate with the shifted test data in our experiments (*e.g.* we do not augment with any of the corruptions in CIFAR-10/100C [18]). An example of a batch we use for adaptation is provided in the supplemental material. Throughout our evaluations, we found that making a small batch from a single image stabilizes the adaptation process and improves the results, although this is not strictly necessary for our adaptation scheme to work. The effect of batches and augmentations is analyzed in our ablation study in Sec. 5.

4. Results

In the following, we evaluate DUA on a variety of tasks and benchmarks. First, we summarize the datasets. Next, we introduce the approaches to which we compare. Lastly, we present our detailed results.



(a) Default Parameters (b) DUA

Figure 3. Running mean and variance of a single channel from the last batch normalization layer for different corruptions in CIFAR-10C. a) Running mean and variance values at each adaptation iteration when default momentum parameters are used. The values are highly unstable which leads to unstable adaptation. b) DUA proposes to use an adaptive momentum schema which leads to fast and stable convergence. This is because of the stability of the running mean and variance values. Initially, the distributions are far apart and thus, we want larger update steps (faster assimilation), whereas later on smaller update steps are beneficial.

	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
Source	67.7	63.1	69.9	55.3	56.6	42.2	50.1	31.6	46.3	39.1	17.1	74.6	34.2	57.9	31.7	49.2
TTT	45.6	41.8	50.0	21.8	46.1	**23.0**	23.9	29.9	30.0	25.1	**12.2**	**23.9**	**22.6**	47.2	**27.2**	31.4
NORM	44.6	43.7	49.1	29.4	45.2	26.2	26.9	25.8	27.9	23.8	18.3	34.3	29.3	37.0	32.5	32.9
DUA	**34.9**	**32.6**	**42.2**	**18.7**	**40.2**	24.0	**18.4**	**23.9**	**24.0**	**20.9**	12.3	27.1	27.2	**26.2**	28.7	**26.8**
Source	28.8	22.9	26.2	9.5	20.6	10.6	9.3	14.2	15.3	17.5	7.6	20.9	14.7	41.3	14.7	18.3
TENT	15.8	13.5	18.7	8.1	18.7	**9.1**	8.0	**10.3**	**10.8**	**11.7**	6.7	11.6	14.1	**11.7**	15.2	12.3
DUA	**15.4**	**13.4**	**17.3**	**8.0**	**18.0**	**9.1**	**7.7**	10.8	**10.8**	12.1	**6.6**	**10.9**	**13.6**	13.0	**14.3**	**12.1**

Table 1. Top-1 Classification Error (%) for each corruption in CIFAR-10C at the highest severity (Level 5). *Source* shows the results from the same model trained on the clean train set and tested on the corrupted test set. For a fair comparison with TTT and NORM, we use ResNet-26 (top), while for TENT, we use the WRN-40-2 (bottom) from their official implementation. Smallest error is shown in bold.

4.1. Benchmarks and Tasks

CIFAR-10/100C: CIFAR-10C and CIFAR-100C [18] are image classification benchmarks to test a model's robustness w.r.t. covariate shifts. These benchmarks add different corruptions to the original test set of CIFAR-10/100 [25] at 5 severity levels. Following the common protocol [42, 60, 63], we evaluate on 15 types of corruptions.

ImageNet-C: Similar to the CIFAR-10/100C benchmarks, ImageNet-C [18] is also an image classification dataset introducing different corruptions at several severity levels to the original test set of ImageNet [7].

KITTI: To test DUA's adaptation capabilities on the task of object detection for autonomous vehicles we use the well-known KITTI [11] dataset. Further, we also use the KITTI-Rain and KITTI-Fog datasets [14] to test the adaptation performance of a KITTI pre-trained model in degrading weather.

4.2. Baselines

We compare our DUA against the following approaches:

- **Source**: denotes the results of the corresponding baseline model trained only on source data, *i.e.* without any adaptation to the test data.
- **TTT**: Test Time Training (TTT) [60] adapts the network parameters by using an auxiliary task on each (out-of-distribution) data sample before testing it.
- **NORM** [42,54]: ignores the train statistics completely and recalculates the batch normalization statistics on the entire test set, leveraging larger batch sizes.

	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
Source	89.5	88.8	95.5	68.4	83.3	65.0	63.5	62.4	74.9	70.3	42.9	83.0	61.1	84.4	65.5	73.2
TTT	83.8	83.0	86.8	59.9	77.7	57.9	59.2	61.5	70.6	70.5	44.5	69.8	56.5	80.2	**60.3**	68.1
NORM	72.5	72.7	77.1	48.6	69.3	**49.7**	47.9	59.5	59.7	58.4	41.8	**53.1**	58.8	57.3	67.7	59.6
DUA	**67.9**	**67.3**	**72.6**	**47.9**	**66.1**	51.6	**46.6**	**58.1**	**57.6**	**54.4**	**41.3**	58.6	**55.3**	**53.3**	60.7	**57.3**
Source	65.7	60.1	59.1	32.0	51.0	33.6	32.4	41.4	45.2	51.4	31.6	55.5	40.3	59.7	42.4	46.7
TENT	**40.3**	**39.9**	41.8	**29.8**	**42.3**	**31.0**	30.0	**34.5**	**35.2**	**39.5**	**28.0**	**33.9**	**38.4**	**33.4**	41.4	**36.0**
DUA	42.2	40.9	**41.0**	30.5	44.8	32.2	**29.9**	38.9	37.2	43.6	29.5	39.2	39.0	35.3	**41.2**	37.6

Table 2. Top-1 Classification Error (%) for each corruption in CIFAR-100C at the highest severity (Level 5).

	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
Source	98.4	97.7	98.4	90.6	93.4	89.8	81.8	89.5	85.0	86.3	51.1	97.2	85.3	76.9	71.7	86.2
TTT	96.9	95.5	96.5	89.9	93.2	86.5	81.5	82.9	82.1	80.0	53.0	**85.6**	79.1	77.2	74.7	83.6
NORM	**87.1**	89.6	90.5	**87.6**	89.4	**80.0**	**71.9**	**70.6**	81.5	**66.9**	47.8	89.8	73.5	**64.2**	68.5	**77.3**
DUA	89.4	**87.6**	**88.1**	88.0	**88.6**	84.7	74.3	77.8	**78.4**	68.6	**45.6**	95.9	**72.2**	66.5	**67.4**	78.2

Table 3. Top-1 Classification Error (%) for each corruption in ImageNet-C at the highest severity (Level 5). *Source* refers to results obtained from a model pre-trained on the original ImageNet and tested on the corrupted test sets. All results are obtained using a ResNet-18 backbone. Smallest error is shown in bold.

- **TENT**: Test time entropy minimization (TENT) [63] recalculates the batch normalization statistics and additionally modifies the scale and shift parameters (γ and β) of the batch normalization layers through back propagation. They obtain the gradients by calculating the prediction entropy on large batches from the out-of-distribution test data.

4.3. Experiments

In this section, we provide a description of all the results obtained on different datasets and benchmarks. We test our DUA during slight distribution shifts and severe domain shifts as well. For our results we always use less than 1% of unlabeled test data and adapt on each incoming sample in a sequential manner (the exact numbers of test samples used for adaptation in our experiments are listed in the supplemental material). Results for all other baselines have been obtained by adapting on the complete test set as stated in their original papers. Note, we also do not need to control for shuffling of the test set in contrast to other approaches [42, 54, 63]. For adaptation, unless stated otherwise, we fix the momentum decay parameter $\omega = 0.94$, and the lower bound $\zeta = 0.005$. Further details for all the experiments are provided in the supplemental material. For reproducibility, code for DUA is available at this repository: `https://github.com/jmiemirza/DUA`

CIFAR-10/100C

For a fair comparison to TTT and NORM we use ResNet-26 [16] and follow the parametrization in their official implementations[1,2]. Similarly, for TENT we use the Wide-ResNet-40-2 [72] from their official implementation[3].

Table 1 shows the results for the highest severity level on CIFAR-10C. Note that we achieve a new state-of-the-art. All other approaches use the complete test set and most also use larger batch sizes and control shuffling of test data. Results on CIFAR-100C are listed in Table 2. Here, DUA outperforms TTT and NORM while being competitive with TENT. Results for lower severity levels are provided in the supplemental material, demonstrating that DUA provides strong results for less severe corruptions as well.

ImageNet-C

For evaluations on ImageNet-C we take an off-the-shelf pre-trained ResNet-18 from PyTorch [44]. Table 3 shows the top-1 error for the highest severity level. DUA performs on-par with all the baselines for ImageNet-C. Results for lower severity levels are provided in the supplemental.

Object Detection

We also test our approach for object detection and show considerable improvement. We conduct these experiments with YOLOv3 [48]. However, our approach could also be applied to other base architectures such as [31, 35, 49, 61], which use batch normalization.

For evaluating our approach on object detection we consider the two following scenarios:

[1]TTT: https://github.com/yueatsprograms/ttt_cifar_release
[2]NORM: https://github.com/bethgelab/robustness
[3]TENT: https://github.com/DequanWang/tent

	Car	Pedestrian	Cyclist
Source only	30.9	34.1	16.2
DUA	51.4	48.5	33.1
Fully Supervised	71.3	64.5	63.2

(a) KITTI → KITTI-Fog

	Car	Pedestrian	Cyclist
Source only	80.7	66.7	54.6
DUA	86.3	70.3	66.7
Fully Supervised	92.3	76.1	78.2

(b) KITTI → KITTI-Rain

Table 4. Results for KITTI pre-trained YOLOv3 tested on rain and fog datasets. We report the Mean Average Precision (mAP@50). a) Results for the most severe fog level, *i.e.* 30m visibility. b) Results for the most severe rain level, *i.e.* 200mm/hr rain intensity.

- Evaluation during covariate shifts; These evaluations are performed to adapt to rain and fog conditions.
- Evaluation during domain shifts; These evaluations test for domain adaptation between datasets.

In degrading weather, a sharp drop in performance of present day object detectors has been noted [40, 41]. Our goal is to dynamically adapt a detector trained on clear weather data to degrading weather conditions.

Adaptation results for the most severe forms of fog and rain augmented on KITTI are shown in Table 4a and 4b, respectively. For fog, the mean improvement across all commonly evaluated classes (*i.e.* car, pedestrian and cyclist) over the source model is 17.7% mAP. Similarly, we also achieve notable improvements while adapting to rain. Here, the mean improvement over the source model is 7.1% mAP. Additional results for varying severity of fog and rain are provided in the supplemental material.

Additional Results

We also demonstrate the benefits of DUA on several other datasets and adaptation tasks in the supplemental material. In particular, we evaluate DUA on:

Digit Recognition: DUA can successfully be used for domain adaptation across datasets which we demonstrate for the task of digit recognition. In particular, we use MNIST [27] and USPS [20], which are datasets consisting of handwritten digits. Additionally, we use SVHN [43], a dataset containing house numbers obtained from Google street view images.

Office-31 [51]**:** is a visual domain adaptation dataset for object classification, containing 31 categories of common objects found in an office environment, captured in three different settings. These include images captured by Webcam, DSLR and gathered from Amazon. We test for domain adaptation across all three settings.

VIS-DA: The Visual Domain Adaptation [45] dataset (VIS-DA) is a large scale image recognition dataset which contains 12 classes. The training set consists of synthetically rendered images. The test set consists of real images cropped from the MS-COCO dataset [32].

SODA10M: The large scale object detection dataset for autonomous vehicles SODA10M [15] provides data captured during day and night. We test for adaptation from day to night. Further, we test for domain adaptation between KITTI and SODA10M.

5. Ablation Studies

In this section we present detailed ablation studies in order to examine our approach more closely.

5.1. Sample Order Does Not Matter

To understand if the ordering of the incoming samples for adaptation holds any significance, we run DUA for 300 independent runs (on CIFAR-10C) and randomly shuffle the test set in each run. The initial, source-only, mean error is 49.2 %. The largest standard deviation over the 300 runs occurs after adaptation on 5 samples, where we achieve 36.4 ± 0.4. Already after 25 samples, we achieve 28.3 ± 0.19. The performance saturates after 100 samples and we achieve 27.2 ± 0.09 (a detailed plot is provided in supplemental material). Thus, the performance of DUA is stable across all independent runs with very little deviation. These results are important to understand that DUA can be used with any arrangement of incoming data.

5.2. Continuous Dynamic Adaptation

In order to understand how good can DUA handle the real-life scenarios where different weather conditions can occur interchangeably, we test DUA for such a scenario on KITTI-Fog: day→fog→day→*etc.* Results are shown in Fig. 4. Note the only minor drop on the source domain (*e.g.* 5.8% worse than the *day baseline* at iteration 100) and that we can quickly recover the same performance again. This shows that despite not being an incremental learning approach, DUA still remembers information from the previous domain. We conjecture that this is because learned weights are not changed during adaptation. DUA only shifts the activation distributions at test time.

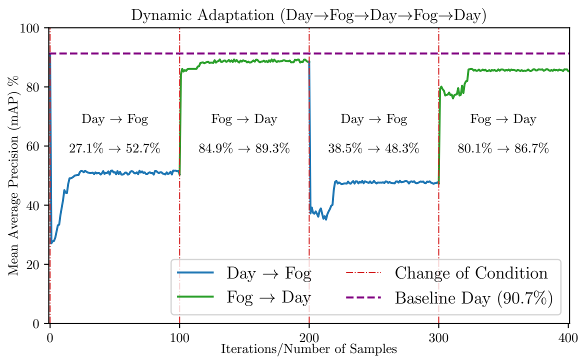


Figure 4. Dynamic adaptation scenario for DUA. We let a KITTI-pretrained model adapt to fog and then back to the original KITTI dataset for two cycles in order to show how well DUA can dynamically adapt to changing weather conditions.

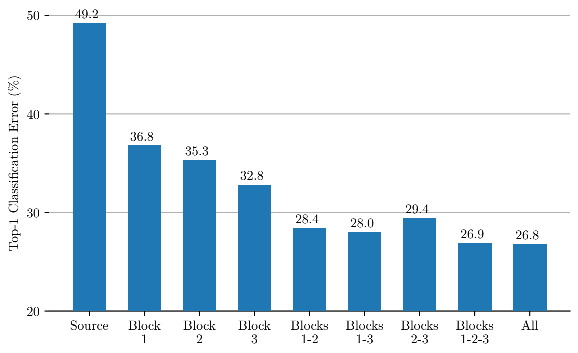


Figure 5. Results on CIFAR-10C after adapting the batch normalization layers of specific ResNet-26 blocks. 'All' refers to adapting all batch normalization layers. This includes the last batch normalization layer after the three ResNet blocks.

5.3. Ablating Batch Normalization Layers

We investigate the effect of adapting only selected batch normalization layers in Figure 5. For this, we adapt batch normalization layers of specific ResNet-26 blocks while keeping all others fixed. As can be seen from the plot, the best performance is obtained by adapting all batch normalization layers in the architecture. Individual improvements are slightly larger at later batch normalization layers.

5.4. Effect of Augmentation

As explained in Section 3.2, we form a small batch of augmented versions from each incoming sample. In Figure 6, we ablate different augmentations and batch sizes to study their effects. Apart from providing stability to our adaptation procedure, making a small batch and augmenting it randomly also provides further improvements. For our experiments, we form a batch of size 64 from each incoming image by augmenting it. However, even a batch size of only 8 suffices to benefit from DUA (with only minor sacrifice in performance).

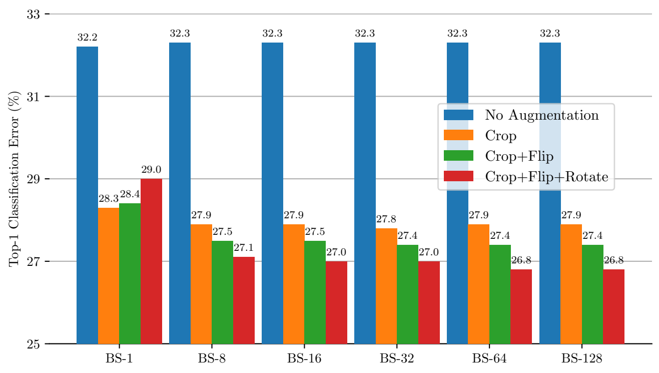


Figure 6. Effects of different batch sizes and augmentations on CIFAR-10C with ResNet-26. DUA makes a batch out of each sequential image by using different random augmentations. Note that the source-only error (without adaptation) is **49.2%**.

6. Conclusion

We have shown that even slight distribution shifts between train and test data can greatly hamper the performance of present day neural networks. We address this limitation by our DUA, which adapts the statistics of a trained model in a sequential manner on each unlabeled sample coming from out-of-distribution test data. To ensure fast and stable adaptation, we introduce an adaptive momentum scheme. DUA does not require access to training data but only needs a fraction of test data to achieve competitive results to strong baselines. Extensive experimentation on a variety of challenging benchmarks and tasks demonstrate the utility of our method on a broad range of batch normalization-based architectures. Since we can dynamically adapt to shifting distributions at a minimal computational overhead, DUA is also well-suited for both real-time systems and embedded devices.

Acknowledgments We gratefully acknowledge the financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association. This work was also partially funded by the Austrian Research Promotion Agency (FFG) under the project High-Scene (884306).

References

[1] Matteo Biasetton, Umberto Michieli, Gianluca Agresti, and Pietro Zanuttigh. Unsupervised Domain Adaptation for Semantic Segmentation of Urban Scenes. In *Proc. CVPRW*, 2019. 2

[2] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. HoMM: Higher-

order Moment Matching for Unsupervised Domain Adaptation. In *Proc. AAAI*, 2020. 2

[3] Hongruixuan Chen, Chen Wu, Yonghao Xu, and Bo Du. Unsupervised Domain Adaptation for Semantic Segmentation via Low-level Edge Information Transfer. *arXiv preprint arXiv:2109.08912*, 2021. 2

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In *Proc. CVPR*, 2018. 1, 2

[5] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No More Discrimination: Cross City Adaptation of Road Scene Segmenters. In *Proc. ICCV*, 2017. 2

[6] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *Proc. CVPR*, 2005. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *Proc. CVPR*, 2009. 5

[8] Christian Fruhwirth-Reisinger, Michael Opitz, Horst Possegger, and Horst Bischof. FAST3D: Flow-Aware Self-Training for 3D Object Detectors. In *Proc. BMVC*, 2021. 2

[9] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch Normalization is a Cause of Adversarial Vulnerability. *arXiv preprint arXiv:1905.02161*, 2019. 3

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *JMLR*, 17(59):1–35, 2016. 1, 2

[11] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *IJR*, 32(11):1231–1237, 2013. 1, 2, 5

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *Proc. ICLR*, 2018. 1, 3

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NeurIPS*, 2014. 2

[14] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based Rendering for Improving Robustness to Rain. In *Proc. ICCV*, 2019. 1, 5

[15] Jianhua Han, Xiwen Liang, Hang Xu, Kai Chen, Lanqing Hong, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Chunjing Xu, and Xiaodan Liang. SODA10M: Towards Large-Scale Object Detection Benchmark for Autonomous Driving. In *NeurIPS*, 2021. 2, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 1, 3, 6

[17] Zhenwei He and Lei Zhang. Multi-adversarial Faster-RCNN for Unrestricted Object Detection. In *Proc. ICCV*, 2019. 1, 2

[18] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. ICLR*, 2019. 1, 4, 5

[19] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional Generative Adversarial Network for Structured Domain Adaptation. In *Proc. CVPR*, 2018. 2

[20] Jonathan J. Hull. A Database for Handwritten Text Recognition Research. *TPAMI*, 16(5):550–554, 1994. 7

[21] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. ICML*, 2015. 2, 3

[22] Maximilian Jaritz, Tuan-Hung Vu, Raoul de Charette, Emilie Wirbel, and Patrick Pérez. xMUDA: Cross-Modal Unsupervised Domain Adaptation for 3D Semantic Segmentation. In *Proc. CVPR*, 2020. 2

[23] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive Adaptation Network for Unsupervised Domain Adaptation. In *Proc. CVPR*, 2019. 2

[24] Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and Match: A Domain Adaptive Representation Learning Paradigm for Object Detection. In *Proc. CVPR*, 2019. 1, 2

[25] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Department of Computer Science, University of Toronto, 2009. 5

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*, 2012. 1

[27] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. GradientBased Learning Applied to Document Recognition. In *Proc. IEEE*, 1998. 7

[28] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting Batch Normalization For Practical Domain Adaptation. In *Proc. ICLRW*, 2016. 2, 3

[29] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing Self-motivated Pyramid Curriculums for Cross-Domain Semantic Segmentation: A Non-Adversarial Approach. In *Proc. ICCV*, 2019. 2

[30] J. Liang et al. Do We Really Need to Access the Source Data? Source Hypothesis Transfer for Unsupervised Domain Adaptation. In *ICML*, 2020. 2

[31] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. In *Proc. ICCV*, 2017. 6

[32] 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 *Proc. ECCV*, 2014. 7

[33] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable Adversarial Training: A General Approach to Adapting Deep Classifiers. In *Proc. ICML*, 2019. 1, 2

[34] Ming-Yu Liu and Oncel Tuzel. Coupled Generative Adversarial Networks. In *NeurIPS*, 2016. 1, 2

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot MultiBox Detector. In *Proc. ECCV*, 2016. 6

[36] Xiaofeng Liu, Site Li, Yubin Ge, Pengyi Ye, Jane You, and Jun Lu. Recursively Conditional Gaussian for Ordinal Unsupervised Domain Adaptation. In *Proc. ICCV*, 2021. 2

[37] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning Transferable Features with Deep Adaptation Networks. In *Proc. ICML*, 2015. 2

[38] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking A Closer Look at Domain Shift: Category-level Adversaries for Semantics Consistent Domain Adaptation. In *Proc. CVPR*, 2019. 1

[39] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. AutoDIAL: Automatic DomaIn Alignment Layers. In *Proc. ICCV*, 2017. 3

[40] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming. *arXiv preprint arXiv:1907.07484*, 2019. 2, 7

[41] Muhammad Jehanzeb Mirza, Cornelius Buerkle, Julio Jarquin, Michael Opitz, Fabian Oboril, Kay-Ulrich Scholl, and Horst Bischof. Robustness of Object Detectors in Degrading Weather Conditions. In *Proc. ITSC*, 2021. 2, 7

[42] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating Prediction-Time Batch Normalization for Robustness under Covariate Shift. *arXiv preprint arXiv:2006.10963*, 2020. 1, 2, 3, 5, 6

[43] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS*, 2011. 7

[44] 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 *NeurIPS*, 2019. 6

[45] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. VisDA: A Synthetic-to-Real Benchmark for Visual Domain Adaptation. In *Proc. CVPRW*, 2017. 7

[46] Can Qin, Haoxuan You, Lichen Wang, C-C Jay Kuo, and Yun Fu. PointDAN: A Multi-Scale 3D Domain Adaption Network for Point Cloud Representation. In *NeurIPS*, 2019. 2

[47] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? In *Proc. ICML*, 2019. 1

[48] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 6

[49] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 6

[50] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986. 2

[51] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting Visual Category Models to New Domains. In *Proc. ECCV*, 2010. 7

[52] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-Weak Distribution Alignment for Adaptive Object Detection. In *Proc. CVPR*, 2019. 1

[53] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Mądry. How Does Batch Normalization Help Optimization? In *NeurIPS*, 2018. 3

[54] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving Robustness Against Common Corruptions by Covariate Shift Adaptation. In *NeurIPS*, 2020. 1, 2, 3, 6

[55] Saurabh Singh and Abhinav Shrivastava. EvalNorm: Estimating Batch Normalization Statistics for Evaluation. In *Proc. ICCV*, 2019. 3

[56] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of Frustratingly Easy Domain Adaptation. In *Proc. AAAI*, 2016. 2

[57] Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *Proc. ECCVW*, 2016. 2

[58] 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 *Proc. CVPR*, 2020. 2

[59] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised Domain Adaptation through Self-Supervision. *arXiv preprint arXiv:1909.11825*, 2019. 2

[60] 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 *Proc. ICML*, 2020. 1, 3, 5

[61] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and Efficient Object Detection. In *Proc. CVPR*, 2020. 6

[62] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial Discriminative Domain Adaptation. In *Proc. CVPR*, 2017. 2

[63] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully Test-time Adaptation by Entropy Minimization. In *Proc. ICLR*, 2020. 2, 3, 5, 6

[64] Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. Cross-domain Contrastive Learning for Unsupervised Domain Adaptation. *arXiv preprint arXiv:2106.05528*, 2021. 2

[65] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in Germany, Test in The USA: Making 3D Object Detectors Generalize. In *Proc. CVPR*, 2020. 2

[66] Yuxin Wu and Justin Johnson. Rethinking "Batch" in BatchNorm. *arXiv preprint arXiv:2105.07576*, 2021. 3

[67] Rongchang Xie, Fei Yu, Jiachao Wang, Yizhou Wang, and Li Zhang. Multi-Level Domain Adaptive Learning for Cross-Domain Detection. In *Proc. ICCVW*, 2019. 1, 2

[68] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *Proc. CVPR*, 2017. 1, 3

[69] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. SPG: Unsupervised Domain Adaptation for 3D Object Detection via Semantic Point Generation. In *Proc. ICCV*, 2021. 2

[70] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proc. CVPR*, 2021. 2

[71] Fei Yu, Mo Zhang, Hexin Dong, Sheng Hu, Bin Dong, and Li Zhang. DAST: Unsupervised Domain Adaptation in Semantic Segmentation Based on Discriminator Attention and Self-Training. In *Proc. AAAI*, 2021. 2

[72] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Proc. BMVC*, 2016. 6

[73] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central Moment Discrepancy (CMD) for Domain-Invariant Representation Learning. In *Proc. ICLR*, 2017. 2

[74] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: Learning to adapt to domain shift. In *NeurIPS*, 2021. 3

[75] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain Object Detection through Coarse-to-Fine Feature Adaptation. In *Proc. CVPR*, 2020. 1, 2

[76] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-Training. In *Proc. ECCV*, 2018. 2