

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

## Supplemental Material

M. Jehanzeb Mirza<sup>1,2</sup>

Jakub Micorek<sup>1</sup>

Horst Possegger<sup>1</sup>

Horst Bischof<sup>1,2</sup>

<sup>1</sup>Institute for Computer Graphics and Vision, Graz University of Technology.

<sup>2</sup>Christian Doppler Laboratory for Embedded Machine Learning.

{muhammad.mirza, jakub.micorek, possegger, bischof}@icg.tugraz.at

In the following, we summarize our evaluation details to support reproducibility (Section 1), provide further ablation results (Section 2) and present detailed results for applying DUA on additional adaptation tasks (Section 3).

## 1. Evaluation Details

**Number of Samples:** DUA requires only a tiny fraction of the (unlabeled) test set to achieve competitively strong results. The exact numbers are listed in Table 1. Note that after adaptation on these specific number of samples, the adaptation performance always saturates. We conducted all experiments on a single NVIDIA® GeForce® RTX 3090.

**Batch Augmentations:** As evaluated in our ablation study (main manuscript Section 5), augmentations help to further improve the adaptation performance. In our experiments we use random cropping, random horizontal flipping and rotating by specific angles (0, 90, 180, 270 degrees). Figure 3 shows an exemplary batch of size 64.

**Corruption Benchmarks:** To train the ResNet-26 [5] backbone on CIFAR-10/100 [8], we use a batch size of 128. We train the model for 150 epochs and use Stochastic Gradient Descent (SGD) as optimizer with learning rate 0.1, momentum 0.9 and weight decay  $5 \cdot 10^{-4}$ . We use the multi-step learning rate scheduler from PyTorch with milestones at 75 and 125 and set its  $\gamma = 0.1$ . For training we use random cropping and horizontal flipping as augmentations.

**Domain Adaptation for Classification:** We use an ImageNet-pretrained ResNet-18 from PyTorch and finetune it on the Office-31 [15] train split. For this finetuning, we train the model for 100 epochs with SGD, initial learning rate 0.1, momentum 0.9 and weight decay  $5 \cdot 10^{-4}$ . We use the PyTorch multi-step learning rate scheduler with milestones at 60 and 75 and set its  $\gamma = 0.1$ . For VIS-DA [12] we use a ResNet-50, while all other settings are kept the

same as for Office-31. We again use random cropping and horizontal flipping as augmentations during training.

**Digit Recognition:** For these experiments we use a simple architecture, consisting of 2 convolution layers, 2 linear layers and 2 batch normalization layers. We also use 2 dropout layers in the classification head for regularization with dropout probabilities  $p = 0.25$  and  $p = 0.5$ , respectively. For each of the datasets, we train this model using ADADELTA [18] for 15 epochs and use a batch size of 64.

**Object Detection:** For all our object detection experiments we use a YOLOv3 [14] pretrained on MS-COCO [10]. Then, we retrain the model for 100 epochs on each of the training splits of the respective datasets. We use a batchsize of 20, while all other optimization settings and augmentation routines for training are taken from the PyTorch implementation<sup>1</sup> of YOLOv3.

## 2. Additional Ablation Results

### 2.1. Sample Order Does Not Matter

As described in the main manuscript (Section 5), the sample order does not matter. Here, we include the detailed plot for all the 300 runs in Figure 1, which shows that DUA is consistently stable across all runs.

### 2.2. Choice of Momentum Decay Parameter

The momentum decay parameter  $\omega$  helps to provide stable and fast adaptation on the incoming samples. The effect of varying  $\omega$  values is analyzed in Figure 2. From this experiment we understand that the adaptation is unstable for a higher value of  $\omega$ , while it is slower for a lower value. Thus, we use  $\omega = 0.94$  for all reported experiments, which empirically provides both stable and fast adaptation.

<sup>1</sup><https://github.com/ultralytics/yolov3>

Dataset	Total no. of samples in test set	No. of samples used for adaptation	% of samples used for adaptation
CIFAR-10C [6]	10000	80	0.8%
CIFAR-100C [6]	10000	80	0.8%
ImageNet-C [6]	50000	100	0.2%
MNIST [9]	10000	30	0.3%
SVHN [11]	26032	30	0.1%
USPS [7]	2007	20	0.9%
Office-31 [15]	4110	40	0.9%
VIS-DA [12]	5534	40	0.7%
KITTI [2]	3741	25	0.6%
KITTI-Rain [3]	3741	25	0.6%
KITTI-Fog [3]	3741	25	0.6%
SODA-10M [4]	4991	30	0.6%
SODA10M-Day [4]	3347	30	0.8%
SODA10M-Night [4]	1644	15	0.9%

Table 1. Available number of samples in the test sets vs. number of samples used for adaptation. Note that all datasets except for KITTI have dedicated test sets. On KITTI, we divide the training set of 7481 images into 3740 train and 3741 test images.

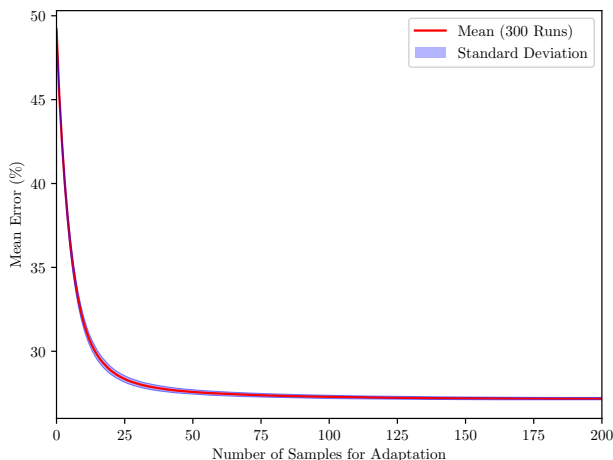


Figure 1. Adaptation results on CIFAR-10C for 300 runs with  $\omega = 0.94$ ,  $\rho_0 = 0.1$ . For each run we randomly shuffle the corrupted test sets. We plot the mean error (on 15 corruptions) and the standard deviation after adaptation on each incoming sample.

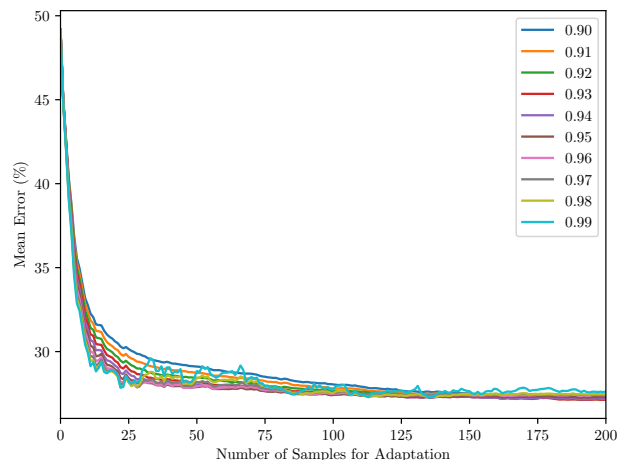


Figure 2. Adaptation results for CIFAR-10C with different values of Momentum Decay Parameter  $\omega$ . At  $\omega = 0.94$ , we achieve a good balance between stable adaptation and fast convergence. For all these experiments  $\rho_0 = 0.1$  is used.

### 3. DUA for Additional Adaptation Tasks

#### 3.1. Domain Adaptation for Digit Recognition

Table 3 summarizes results for cross-dataset domain adaptation in digit recognition. For all these experiments we use a simple architecture consisting of 2 convolution layers, 2 batch normalization layers and 2 fully connected layers. Our method improves the results for all the popular domain adaptation benchmarks. Note that we only use random cropping for digit recognition experiments.

#### 3.2. Domain Adaptation for Visual Recognition

In Table 4, we list the results obtained by testing on two popular domain adaptation benchmarks, Office-31 and VIS-DA. For VIS-DA we take a ResNet-50, pre-trained on ImageNet and then fine tune it on the VIS-DA train split. For the Office-31 dataset, we use an ImageNet pre-trained ResNet-18 which we finetune on the Office-31 train split.

We show that our method achieves improvements on both Office-31 and VIS-DA datasets. It should be noted that the purpose of these evaluations is not to outperform the state-of-the-art methods which specialize on these tasks. Instead, we show that DUA is applicable to a variety of tasks

Corruption Type	Abbreviation
Gaussian Noise	gaus
Shot Noise	shot
Impulse Noise	impul
Defocus Blur	defcs
Glass Blur	gls
Motion Blur	mtn
Zoom Blur	zm
Snow	snw
Frost	frst
Fog	fg
Brightness	brt
Contrast	cnt
Elastic	els
Pixelate	px
JPEG Compression	jpg

Table 2. Abbreviations of corruption types in CIFAR-10/100C and ImageNet-C.

and architectures. It can be used as an initial adaptation method before applying other established domain adaptation methods such as [1, 13, 16, 17].

### 3.3. Corruption Benchmarks

We provide detailed results for the lower severity corruption levels 1–4 (highest severity level 5 is included in the main manuscript) for CIFAR-10C (Table 5), CIFAR-100C (Table 6) and ImageNet-C (Table 7). Note that the abbreviations of the different corruption types are summarized in Table 2. As can be seen from these results, our DUA achieves consistent improvements over all corruption types and severity levels.

### 3.4. Object Detection

**Natural Domain Shifts:** Table 8a shows the results for adapting a detector pre-trained on day images only and tested on night images. Table 8b and Table 8c report the domain adaptation results between KITTI and SODA10M. DUA provides notable gains for all the different scenarios.

**Degrading Weather:** We provide results for the lower severity levels of KITTI-Fog and KITTI-Rain [3] in Table 9 (highest severity/lowest visibility is included in the main manuscript). We see that by adapting a model with DUA, the detection performance increases even in lower severities of rain and fog.

## References

- [1] 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. 3
- [2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets Robotics: The KITTI Dataset. *IJR*, 32(11):1231–1237, 2013. 2, 7
- [3] Shirsendu Sukanta Halder, Jean-François Lalonde, and Raoul de Charette. Physics-based Rendering for Improving Robustness to Rain. In *Proc. ICCV*, 2019. 2, 3, 7
- [4] 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
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016. 1
- [6] Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proc. ICLR*, 2019. 2, 4
- [7] Jonathan J. Hull. A Database for Handwritten Text Recognition Research. *TPAMI*, 16(5):550–554, 1994. 2, 4
- [8] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, Department of Computer Science, University of Toronto, 2009. 1
- [9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. GradientBased Learning Applied to Document Recognition. In *Proc. IEEE*, 1998. 2, 4
- [10] 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. 1
- [11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bis-sacco, Bo Wu, and Andrew Y Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NeurIPS*, 2011. 2, 4
- [12] 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. 1, 2, 4
- [13] Pedro O Pinheiro. Unsupervised Domain Adaptation with Similarity Learning. In *Proc. CVPR*, 2018. 3
- [14] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1
- [15] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting Visual Category Models to New Domains. In *Proc. ECCV*, 2010. 1, 2, 4
- [16] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of Frustratingly Easy Domain Adaptation. In *Proc. AAAI*, 2016. 3
- [17] 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. 3
- [18] Matthew D Zeiler. ADADELTA: An Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*, 2012. 1










Source data:	SVHN [11]	SVHN	MNIST [9]	MNIST	USPS [7]	USPS	Exemplary Images		
Target data:	USPS	MNIST	USPS	SVHN	MNIST	SVHN	MNIST	SVHN	USPS
Source only	66	58	78	16	42	9			
DUA	71	68	86	33	55	28			
Fully Supervised	94	96	95	85	97	91			

Table 3. Domain adaptation results (measured by accuracy in %) for digit recognition. All results are obtained by using a simple architecture consisting of 2 convolution layers, 2 fully connected layers and batch normalization layers.

Source data:	Amazon	Amazon	DSLR	DSLR	Webcam	Webcam		
Target data:	Webcam	DSLR	Webcam	Amazon	DSLR	Amazon		Classification Error (%)
Source only	38.4	34.3	12.9	61.4	4.2	66.1	Source only	57.4
DUA	33.2	29.6	5.7	51.9	2.0	59.7	DUA	47.2
Fully Supervised	27.1	11.3	3.9	34.1	1.2	22.3	Fully Supervised	23.7

(a) Office-31 [15]

(b) VIS-DA [12]

Table 4. Domain adaptation results (measured by classification error in %) for a) Office-31 using a ResNet-18 backbone, and for b) VIS-DA using a ResNet-50 backbone.

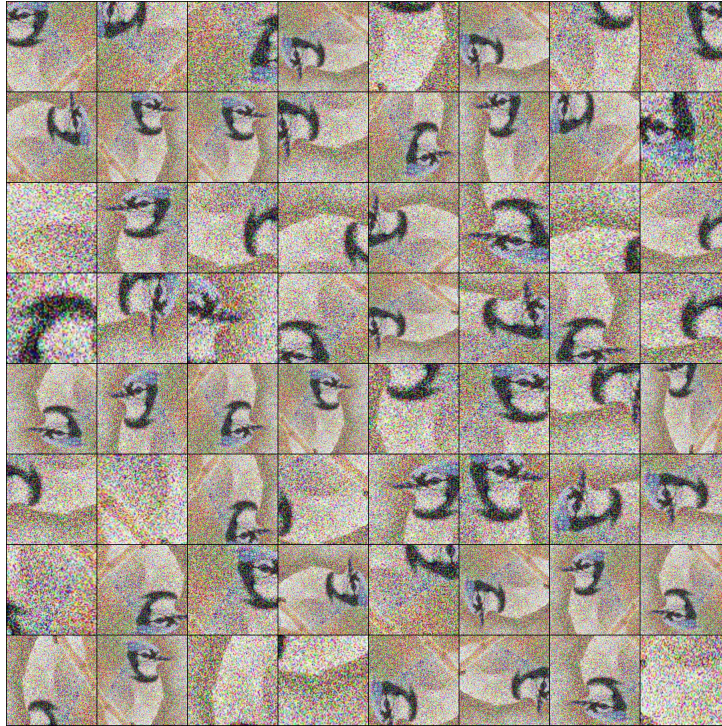


Figure 3. Example of a batch which we create from a single image by augmenting it randomly. The input sample is taken from ImageNet-C [6] (Level 5) Gaussian Noise Corruption.



	gaus	shot	impul	defcs	glc	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
<b>Level 4</b>																
Source	63.9	53.7	57.0	28.9	58.9	32.4	38.1	25.9	33.9	17.5	10.4	33.7	26.7	40.7	27.2	36.6
TTT	41.5	35.4	39.8	15.0	47.8	<b>19.1</b>	18.4	<b>20.1</b>	24.0	13.5	<b>10.0</b>	<b>14.1</b>	<b>17.7</b>	29.4	<b>24.5</b>	24.7
NORM	40.7	37.4	43.2	16.7	47.4	21.8	20.2	29.9	30.3	19.0	16.1	20.5	26.5	26.6	35.7	28.8
DUA	<b>31.0</b>	<b>27.6</b>	<b>35.8</b>	<b>13.2</b>	<b>40.7</b>	20.3	<b>15.4</b>	22.2	<b>20.6</b>	<b>12.7</b>	10.1	14.8	20.5	<b>18.6</b>	24.6	<b>21.9</b>
Source	24.1	17.1	16.4	6.6	23.5	8.4	7.4	12.2	11.5	8.3	6.2	9.2	10.6	19.4	13.1	12.9
TENT	13.8	<b>11.7</b>	14.3	6.7	18.6	8.2	7.1	10.6	9.7	7.5	6.1	8.4	10.9	<b>8.5</b>	13.2	10.3
DUA	<b>13.7</b>	11.8	<b>13.5</b>	<b>5.9</b>	<b>18.3</b>	<b>7.6</b>	<b>6.6</b>	<b>10.3</b>	<b>9.0</b>	<b>7.4</b>	<b>5.8</b>	<b>7.2</b>	<b>9.9</b>	9.3	<b>13.0</b>	<b>10.0</b>
<b>Level 3</b>																
Source	58.0	47.5	38.5	17.7	46.2	32.8	30.6	22.7	31.8	12.6	9.5	19.3	20.7	23.7	24.7	29.1
TTT	37.2	31.6	28.6	11.5	35.8	<b>19.1</b>	15.8	<b>17.8</b>	23.3	<b>11.0</b>	<b>9.1</b>	<b>11.6</b>	<b>14.3</b>	18.9	<b>22.3</b>	20.5
NORM	37.8	35.1	34.7	14.1	38.2	21.7	18.2	27.5	29.0	16.6	15.2	18.6	19.6	21.1	33.3	25.4
DUA	28.3	<b>24.6</b>	<b>27.0</b>	<b>10.4</b>	<b>30.7</b>	20.2	<b>14.4</b>	20.4	<b>19.3</b>	<b>11.0</b>	9.2	12.3	14.6	<b>15.1</b>	23.1	<b>18.7</b>
Source	20.4	14.6	9.7	<b>5.4</b>	12.9	8.6	6.5	9.9	11.4	6.3	5.5	7.2	7.4	9.6	12.1	9.8
TENT	12.6	<b>10.4</b>	10.4	6.0	12.7	8.1	6.7	9.5	<b>9.1</b>	6.7	5.9	7.5	8.4	7.5	12.7	8.9
DUA	<b>12.2</b>	10.5	<b>9.3</b>	5.5	<b>11.9</b>	<b>7.8</b>	<b>6.1</b>	<b>9.1</b>	<b>9.1</b>	<b>6.1</b>	<b>5.4</b>	<b>6.3</b>	<b>7.0</b>	<b>7.4</b>	<b>11.7</b>	<b>8.4</b>
<b>Level 2</b>																
Source	43.1	27.8	29.3	10.2	49.5	23.4	22.4	26.4	21.3	10.3	8.7	13.4	14.7	17.9	22.3	22.7
TTT	28.8	20.7	23.0	<b>9.0</b>	36.6	<b>15.4</b>	13.1	<b>20.2</b>	16.9	<b>9.2</b>	<b>8.3</b>	<b>10.2</b>	<b>12.5</b>	14.8	<b>19.7</b>	17.2
NORM	31.0	25.3	28.7	13.5	38.8	18.8	16.3	27.8	23.9	15.4	14.6	17.1	18.7	19.6	30.6	22.7
DUA	<b>22.3</b>	<b>16.8</b>	<b>22.9</b>	9.2	<b>30.3</b>	16.0	<b>12.7</b>	21.5	<b>15.7</b>	9.6	8.7	11.1	12.7	<b>13.3</b>	20.8	<b>16.2</b>
Source	13.4	8.8	8.0	<b>5.1</b>	14.2	6.5	5.8	9.2	8.5	5.3	5.3	6.1	6.5	7.8	<b>10.9</b>	8.1
TENT	10.2	7.6	8.6	5.9	13.0	7.2	6.2	<b>8.1</b>	7.8	6.3	5.8	6.9	7.5	7.0	11.8	8.0
DUA	<b>10.0</b>	<b>7.5</b>	<b>7.6</b>	<b>5.1</b>	<b>12.4</b>	<b>6.4</b>	<b>5.7</b>	8.3	<b>7.3</b>	<b>5.2</b>	<b>5.2</b>	<b>5.7</b>	<b>6.4</b>	<b>6.8</b>	<b>10.9</b>	<b>7.4</b>
<b>Level 1</b>																
Source	25.8	18.4	19.0	8.5	51.1	14.7	18.2	15.0	13.8	8.3	8.3	8.7	14.4	11.3	16.5	16.8
TTT	19.1	15.8	<b>16.5</b>	<b>8.0</b>	37.9	<b>11.7</b>	<b>12.2</b>	<b>12.8</b>	<b>11.9</b>	<b>8.2</b>	<b>8.0</b>	<b>8.3</b>	<b>12.6</b>	11.1	<b>15.5</b>	14.0
NORM	24.0	20.9	22.5	13.4	38.1	16.5	15.5	20.5	18.8	14.9	14.0	15.3	19.1	16.9	24.7	19.7
DUA	<b>16.5</b>	<b>13.8</b>	16.6	8.3	<b>30.4</b>	12.4	12.6	14.5	12.2	8.4	8.4	8.8	13.4	<b>11.0</b>	15.9	<b>13.6</b>
Source	8.7	6.5	<b>6.2</b>	<b>4.9</b>	14.1	<b>5.5</b>	5.9	6.4	6.5	<b>4.9</b>	<b>5.0</b>	<b>5.0</b>	<b>6.9</b>	<b>5.8</b>	8.7	6.7
TENT	7.5	6.9	7.2	5.7	12.4	6.2	6.3	6.8	6.5	5.9	5.7	6.0	7.9	6.5	9.1	7.1
DUA	<b>7.3</b>	<b>6.2</b>	<b>6.2</b>	5.1	<b>11.9</b>	<b>5.5</b>	<b>5.8</b>	<b>6.2</b>	<b>6.1</b>	5.1	5.1	5.1	7.0	<b>5.8</b>	<b>8.5</b>	<b>6.5</b>

Table 5. Error (%) for each corruption in CIFAR-10C severity (Level 1–4) is reported. Source refers to results obtained from a model trained on clean train set and tested on corrupted test sets. For a fair comparison with TTT and NORM, ResNet-26 is used. For comparison with TENT we take the Wide-ResNet-40-2 model from their official Github Repository. Lowest error is highlighted for each corruption.

	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
<b>Level 4</b>																
Source	88.3	85.3	92.8	54.3	84.8	57.3	57.2	54.9	65.9	48.6	<b>35.8</b>	61.1	52.9	73.0	61.0	64.9
TTT	81.6	78.3	81.1	48.6	78.7	52.5	53.4	53.8	62.8	49.5	38.5	50.3	50.2	66.2	<b>57.2</b>	60.2
NORM	70.5	67.6	72.1	<b>41.1</b>	69.9	<b>46.1</b>	44.5	54.7	55.2	46.7	38.8	<b>44.1</b>	50.8	49.9	64.2	54.4
DUA	<b>66.0</b>	<b>62.9</b>	<b>66.4</b>	41.3	<b>66.5</b>	48.2	<b>43.0</b>	<b>53.5</b>	<b>52.8</b>	<b>43.2</b>	37.6	45.7	<b>48.2</b>	<b>46.6</b>	57.6	<b>52.0</b>
Source	60.7	51.6	47.9	27.1	54.4	30.3	28.9	37.4	39.0	35.4	27.2	35.9	34.4	39.0	40.1	39.3
TENT	<b>38.9</b>	<b>36.3</b>	<b>36.6</b>	27.3	<b>42.0</b>	<b>28.9</b>	28.4	<b>34.8</b>	<b>32.8</b>	<b>32.1</b>	<b>26.3</b>	<b>29.8</b>	<b>33.3</b>	<b>29.9</b>	<b>38.4</b>	<b>33.1</b>
DUA	43.0	39.0	37.9	<b>26.9</b>	44.7	29.5	<b>28.3</b>	36.5	34.3	33.8	26.6	31.0	33.8	31.0	38.9	34.3
<b>Level 3</b>																
Source	86.0	81.4	83.8	42.7	78.3	57.7	52.4	53.0	64.1	40.1	<b>33.3</b>	49.3	46.6	54.4	58.2	58.8
TTT	79.6	74.6	69.3	42.5	73.0	53.2	49.8	51.2	61.4	42.1	36.8	43.5	45.8	52.9	<b>55.2</b>	55.4
NORM	67.7	64.6	62.3	60.9	63.4	<b>42.6</b>	52.6	54.5	<b>41.5</b>	<b>37.8</b>	37.8	<b>41.1</b>	44.3	45.0	61.7	51.9
DUA	<b>63.8</b>	<b>60.0</b>	<b>57.9</b>	<b>37.3</b>	<b>58.4</b>	48.2	<b>41.2</b>	<b>50.8</b>	52.4	39.2	35.5	41.4	<b>41.7</b>	<b>42.1</b>	55.3	<b>48.4</b>
Source	55.2	45.9	36.9	25.7	39.9	30.5	27.4	33.3	38.1	29.5	25.5	30.5	28.6	30.3	38.0	34.4
TENT	<b>37.1</b>	<b>34.3</b>	<b>32.0</b>	26.1	<b>34.6</b>	<b>29.2</b>	27.7	<b>32.3</b>	<b>32.3</b>	<b>29.0</b>	25.6	<b>28.4</b>	29.2	<b>28.3</b>	37.4	<b>30.9</b>
DUA	40.8	36.5	32.2	<b>25.3</b>	37.0	29.6	<b>27.1</b>	32.7	34.4	<b>29.0</b>	<b>25.2</b>	28.6	<b>28.4</b>	28.7	<b>37.3</b>	31.5
<b>Level 2</b>																
Source	79.9	66.1	73.7	<b>33.7</b>	79.6	48.9	46.4	56.8	52.4	<b>35.2</b>	<b>31.3</b>	42.1	40.5	48.0	55.2	52.7
TTT	71.5	61.8	59.8	36.5	73.1	47.2	46.0	55.7	52.8	38.0	35.2	39.7	42.0	47.9	<b>52.6</b>	50.7
NORM	61.1	54.9	55.9	36.3	61.1	<b>43.0</b>	40.7	53.0	49.0	39.6	37.0	39.5	42.8	43.3	59.0	47.7
DUA	<b>57.1</b>	<b>50.7</b>	<b>52.0</b>	35.2	<b>58.4</b>	43.8	<b>39.1</b>	<b>52.8</b>	<b>47.6</b>	35.9	34.2	<b>38.9</b>	<b>39.9</b>	<b>39.7</b>	52.9	<b>45.2</b>
Source	44.6	34.5	30.7	24.3	41.5	27.7	26.2	32.7	31.8	26.8	24.4	27.5	<b>27.9</b>	28.0	36.5	31.0
TENT	<b>33.3</b>	<b>29.5</b>	<b>28.8</b>	25.7	<b>34.9</b>	27.4	27.0	<b>30.6</b>	<b>29.6</b>	27.0	25.4	27.3	29.1	<b>27.6</b>	36.3	<b>29.3</b>
DUA	35.9	31.2	<b>28.8</b>	<b>24.1</b>	36.9	<b>27.2</b>	<b>26.0</b>	32.2	30.9	<b>26.1</b>	<b>24.0</b>	<b>26.6</b>	<b>27.9</b>	<b>27.6</b>	<b>35.9</b>	29.4
<b>Level 1</b>																
Source	65.0	53.4	54.2	<b>30.3</b>	80.1	40.2	42.9	<b>39.6</b>	43.0	<b>31.0</b>	<b>30.2</b>	<b>31.8</b>	<b>40.3</b>	37.0	48.1	44.5
TTT	60.4	53.0	48.0	34.7	74.0	41.3	41.3	41.5	44.2	34.6	34.4	34.8	41.8	39.4	<b>47.0</b>	44.7
NORM	52.9	48.8	47.7	36.2	60.6	40.1	39.5	43.9	44.0	36.8	36.3	36.7	43.2	40.6	52.0	44.0
DUA	<b>49.6</b>	<b>45.5</b>	<b>44.4</b>	32.5	<b>58.0</b>	<b>39.4</b>	<b>38.3</b>	41.7	<b>41.9</b>	33.2	33.1	33.5	40.8	<b>36.0</b>	<b>47.0</b>	<b>41.0</b>
Source	34.4	29.6	26.9	<b>23.8</b>	42.9	25.6	26.1	<b>26.1</b>	27.4	<b>24.0</b>	<b>23.8</b>	24.3	<b>28.4</b>	<b>25.2</b>	32.4	28.1
TENT	<b>29.3</b>	<b>27.6</b>	26.9	25.5	<b>34.7</b>	26.6	26.7	27.0	27.3	25.6	25.2	26.0	29.8	26.7	32.9	27.9
DUA	31.2	28.5	<b>26.4</b>	<b>23.8</b>	36.9	<b>25.3</b>	<b>25.9</b>	26.2	<b>27.1</b>	24.1	23.9	<b>23.9</b>	28.6	25.4	<b>32.0</b>	<b>27.3</b>

Table 6. Error (%) for each corruption in CIFAR-100C severity (Level 1–4) is reported. Source refers to results obtained from a model trained on clean train set and tested on corrupted test sets. For a fair comparison with TTT and NORM, ResNet-26 is used. For comparison with TENT we take the Wide-ResNet-40-2 model from their official Github Repository. Lowest error is highlighted for each corruption.

	gaus	shot	impul	defcs	gls	mtn	zm	snw	frst	fg	brt	cnt	els	px	jpg	mean
<b>Level 4</b>																
Source	93.2	94.7	94.3	84.5	89.4	85.3	77.2	83.4	79.4	72.8	44.5	88.1	63.4	71.2	58.8	78.7
NORM	84.4	<b>77.6</b>	87.3	86.4	88.0	<b>76.8</b>	70.7	76.9	<b>70.9</b>	<b>56.0</b>	<b>37.8</b>	<b>64.4</b>	<b>53.5</b>	<b>58.6</b>	57.8	<b>69.8</b>
DUA	<b>78.1</b>	82.8	<b>80.3</b>	<b>82.8</b>	<b>83.3</b>	78.7	<b>69.8</b>	<b>76.1</b>	74.2	59.7	40.4	87.3	55.8	61.8	<b>54.9</b>	71.1
<b>Level 3</b>																
Source	80.9	82.7	82.9	74.1	85.4	73.9	71.9	73.6	77.8	66.2	39.6	65.3	51.1	56.7	49.3	68.8
NORM	<b>65.5</b>	69.8	<b>63.2</b>	71.1	79.5	<b>63.6</b>	<b>58.2</b>	68.1	<b>65.4</b>	55.3	39.3	<b>55.0</b>	<b>41.0</b>	51.8	50.9	<b>59.8</b>
DUA	67.6	<b>68.5</b>	69.0	<b>70.7</b>	<b>78.6</b>	66.4	65.3	<b>66.9</b>	70.6	<b>54.3</b>	<b>37.4</b>	61.8	46.5	<b>47.6</b>	<b>48.2</b>	61.3
<b>Level 2</b>																
Source	63.6	67.8	74.6	59.4	65.4	57.7	65.2	77.5	67.0	56.3	36.4	51.5	60.9	43.3	46.3	59.5
NORM	<b>50.7</b>	61.2	65.2	61.3	<b>54.4</b>	<b>46.4</b>	<b>55.8</b>	<b>69.1</b>	<b>58.0</b>	<b>45.6</b>	36.1	<b>40.3</b>	64.1	41.3	<b>40.6</b>	<b>52.7</b>
DUA	56.4	<b>58.0</b>	<b>62.0</b>	<b>56.5</b>	62.0	53.1	58.0	69.9	63.0	49.3	<b>35.8</b>	50.5	<b>56.6</b>	<b>40.6</b>	45.3	54.5
<b>Level 1</b>																
Source	50.5	53.1	61.9	51.3	52.4	45.2	55.9	55.5	49.7	49.0	34.2	44.0	40.4	41.3	42.6	48.5
NORM	47.0	<b>48.4</b>	<b>54.1</b>	<b>50.2</b>	<b>47.8</b>	<b>39.5</b>	50.3	<b>47.7</b>	<b>44.1</b>	<b>41.1</b>	38.4	<b>36.6</b>	42.0	<b>37.0</b>	43.2	<b>44.5</b>
DUA	<b>45.7</b>	49.0	54.7	50.6	49.8	43.4	<b>50.1</b>	51.7	46.9	45.8	<b>31.8</b>	43.3	<b>38.9</b>	39.1	<b>41.5</b>	45.5

Table 7. Error (%) for each corruption in ImageNet-C severity (Level 1–4) is reported. Source refers to results obtained from a model pre-trained on ImageNet and tested on corrupted test sets. All results are obtained by using ResNet-18 as the backbone architecture. Lowest error is highlighted for each corruption.

	Car	Ped	Cyclist		Car	Ped	Cyclist		Car	Ped	Cyclist
SO	75.3	48.3	50.1	SO	43.6	13.9	18.9	SO	80.1	55.5	33.9
DUA	77.2	49.9	51.6	DUA	49.4	17.2	23.1	DUA	81.3	56.3	35.1
FS	86.9	55.7	63.2	FS	84.8	46.1	61.2	FS	91.8	71.3	76.5

(a) SODA10M [4] (Day → Night)      (b) KITTI [2] → SODA10M      (c) SODA10M → KITTI

Table 8. Domain adaptation results (mAP@50) for object detection with YOLOv3. a) We train on the SODA10M day split and test on the SODA10M night split. b) We train on the KITTI train split and test on the SODA10M test split. c) We train on the SODA10M day split and test on the KITTI validation split (details in Table 1). SO: source-only, FS: fully supervised.

	Car	Pedestrian	Cyclist		Car	Pedestrian	Cyclist
40m fog visibility				100 mm/hr rain			
Source only	39.6	39.3	20.2	Source only	93.2	80.5	81.1
DUA	61.3	53.5	39.9	DUA	94.9	81.7	83.2
Fully Supervised	77.8	68.1	70.5	Fully Supervised	95.4	83.2	84.1
50m fog visibility				75 mm/hr rain			
Source only	48.2	45.9	27.3	Source only	95.5	82.6	83.6
DUA	69.6	59.7	48.4	DUA	96.1	83.2	86.0
Fully Supervised	82.2	71.3	74.2	Fully Supervised	96.6	84.1	87.2

(a) KITTI → KITTI-Fog [3]      (b) KITTI → KITTI-Rain [3]

Table 9. Results for KITTI pretrained YOLOv3 tested on rain and fog datasets. Mean Average Precision (mAP@50) is reported for the three common classes in KITTI dataset. a) Results for 40m and 50m visibility in fog. b) Results for 100mm/hr and 75mm/hr rain intensity.