

Supplementary: AR-TTA: A Simple Method for Real-World Continual Test-Time Adaptation

Damian Sójka^{1,2}

damian.sojka@doctorate.put.poznan.pl

Bartłomiej Twardowski^{1,3}

btwardowski@cvc.uab.es

Tomasz Trzciński^{1,4,5}

tomasz.trzcinski@pw.edu.pl

Sebastian Cygert^{1,6}

sebcyg@multimed.org

¹ IDEAS-NCBR, Warsaw, Poland

² Institute of Robotics and Machine Intelligence

Poznan University of Technology
Poznań, Poland

³ Computer Vision Center
Universitat Autònoma de Barcelona
Barcelona, Spain

⁴ Warsaw University of Technology
Warsaw, Poland

⁵ Tooplox, Wrocław, Poland

⁶ Gdańsk University of Technology
Gdańsk, Poland

This document provides more results and describes our experimental procedure in detail. Section 1 provides a detailed analysis of our method: effect of memory replay, adapting different model layers, and hyper-parameter analysis. Section 2 provides detailed results of the experiments from the main paper. Section 3 described implementation details including the hyper-parameter search for all of the methods. Section 4 described benchmarks used in this paper, including the introduced ones SHIFT-C and CLAD-C.

1 AR-TTA detailed analysis

1.1 Effect Of Replay Memory Size

The necessity to keep a set of samples from source data in memory can be problematic in memory-limited settings. We verified the possibility of minimizing the size of replay memory and evaluated our method with different numbers of stored samples. The results in Figure 1 show that our method is robust to replay memory size. There is no significant difference in accuracy between memory sizes of 500 and 10000 for both CIFAR10C and CLAD-C benchmarks. A slight degradation in performance can be seen with only 100 exemplars for CIFAR10C. Less severe domain shift in CLAD-C allows for a more significant reduction in the memory size without the performance drop.

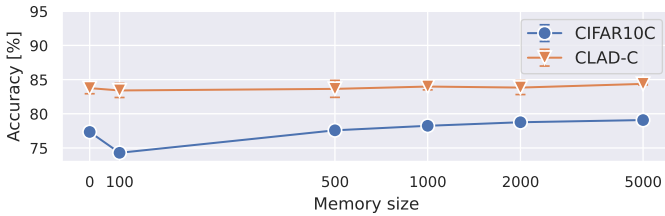


Figure 1: The influence of replay memory size on the resulting accuracy on CIFAR10C and CLAD-C benchmarks.

1.2 Adapted Weights

Table 1 shows the performance related to different configurations of adapted weights with our proposed method. We check the multiple configurations of adapting the last two layers, the first two layers, and only BN statistics. The best results are achieved by adapting the whole model.

Table 1: Classification accuracy (%) for different configurations of adapted weights with our proposed method AR-TTA.

CIFAR10C (WideResNet28)		CLAD-C (ResNet50)	
Adapted weights	Mean	Adapted weights	Mean
block 1 (BN affine only)	61.4	layer 1 (BN affine only)	81.7
block 1, 2 (BN affine only)	26.3	layer 1, 2 (BN affine only)	81.5
block 2, 3 (BN affine only)	73.8	layer 3, 4 (BN affine only)	81.9
block 3 (BN affine only)	72.8	layer 4 (BN affine only)	82.1
block 1	17.4	layer 1	80.4
block 1, 2	13.9	layer 1, 2	81.6
block 2, 3	78.4	layer 3, 4	83.3
block 3	77.2	layer 4	82.8
BN affine only	75.1	BN affine only	81.8
Whole model (Ours)	78.8	Whole model (Ours)	83.7

1.3 Influence Of Beta Distribution Shape For Mixup Augmentation

The beta distribution in mixup augmentation is used to sample interpolation parameter between exemplars. Within our method, it controls the interpolation between test data samples and exemplars from the replay memory bank. By shaping this distribution we can adjust what are the fractions of replay and test data in the augmented samples. Results are shown in Table 2. The shape of the distribution did not have a significant impact on the results. The symmetric shape of the distribution, common for mixup augmentation, gives the best results.

Table 2: Classification accuracy (%) for CIFAR10C and CLAD-C tasks for different configurations of beta distribution parameters ψ and ρ for sampling interpolation parameter $\lambda \sim \text{Beta}(\psi, \rho)$ required for mixup data augmentation.

ψ	ρ	CIFAR10C	CLAD-C
5.0	5.0	78.6	83.7
1.0	5.0	78.6	81.8
5.0	1.0	78.2	83.0
2.0	8.0	74.8	82.0
8.0	2.0	77.5	83.8
Ours			
0.4	0.4	78.8	83.7

1.4 Additional Component Analysis

Table 3 shows results for different component configurations of our method. It includes the experiment without the usage of a weight-averaged teacher. We utilized pseudo-labels from the adapted model itself (configuration A). Additionally, we show the performance of our method when chosen exemplars for replay memory are not class-balanced.

Table 3: Classification accuracy (%) for CIFAR10C and CLAD-C tasks for different configurations of the proposed method.

Method	CIFAR10C	CLAD-C
A: Pseudo-labels	75.5 \pm 0.07	71.3 \pm 0.54
B: A + Weight-avg. teacher	75.7 \pm 0.07	71.1 \pm 0.53
C: B + Replay memory	77.3 \pm 0.16	69.0 \pm 0.66
D: C + Mixup	78.5 \pm 0.13	72.2 \pm 0.31
E: B + Dynamic BN stats	77.3 \pm 0.07	83.8 \pm 0.82
F: E + Replay memory	79.8 \pm 0.03	82.8 \pm 1.09
AR-TTA (Ours) with random memory selection	77.1 \pm 0.36	83.7 \pm 0.81
AR-TTA (Ours)	78.8 \pm 0.13	83.7 \pm 0.64

1.5 Dynamic Batch Norm Statistics Analysis

The γ is a scale parameter of the distance between distributions $D(\phi^S, \phi_t^T)$. It determines the magnitude of the calculated values of β , which is used for linear interpolation between the saved source batch normalization (BN) statistics ϕ^S and the BN statistics calculated from the current batch ϕ_t^T . The higher the value of γ , the higher the values of β tend to be. At the same time, the higher the β values, the more influence BN statistics from current batch have on interpolation and calculation of the finally used BN statistics. In Figure 2 we show the relationship between γ parameter value and mean accuracy of our AR-TTA method for CIFAR10-to-CIFAR10C and CLAD-C benchmarks. We can see the contradicting trend between the two benchmarks. This suggests that the discrepancy in the data distribution between the source domain and the estimated distribution for each test data batch is more prominent in CIFAR10C compared to CLAD-C. This is in agreement with the results of the BN-1 [14] baseline method. BN-1 discards the BN statistics from the source data. Its performance was significantly better on CIFAR10C and worse on CLAD-C, compared to the fixed source model.

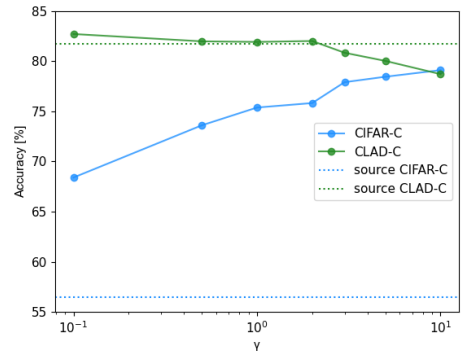


Figure 2: The relationship between mean classification accuracy (%) and the value of parameter γ for CIFAR10C and CLAD-C benchmarks.

2 Additional Results

We present batch-wise accuracy plots for the CIFAR10C, ImageNet-C, CIFAR10.1, and SHIFT-C benchmarks in Figures 3, 4, 5, 6, respectively. Moreover, the full results on CIFAR10C can be found in Table 4.

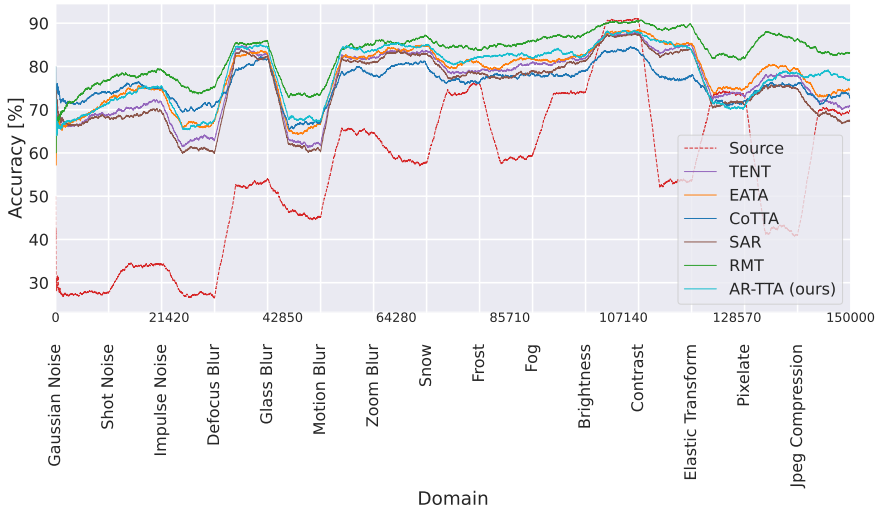


Figure 3: Batch-wise classification accuracy (%) averaged in a window of 400 batches on CIFAR10C benchmark for the chosen methods continually adapted to the sequences of data. The major ticks on the x-axis symbolize the beginning of the next sequence and, at the same time, a different domain. Minor ticks on the x-axis (numbers) indicate the image number. Best viewed in color.

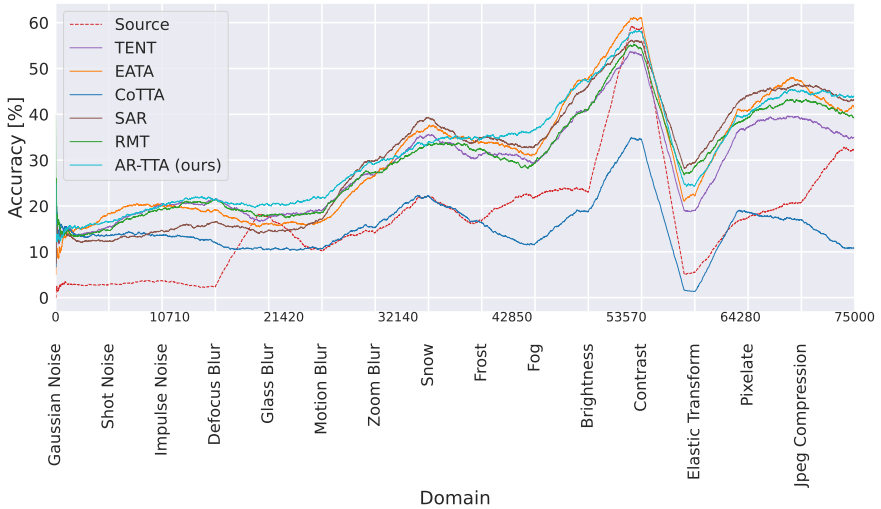


Figure 4: Batch-wise classification accuracy (%) averaged in a window of 400 batches on ImageNet-C benchmark for the chosen methods continually adapted to the sequences of data. The major ticks on the x-axis symbolize the beginning of the next sequence and, at the same time, a different domain. Minor ticks on the x-axis (numbers) indicate the image number. Best viewed in color.

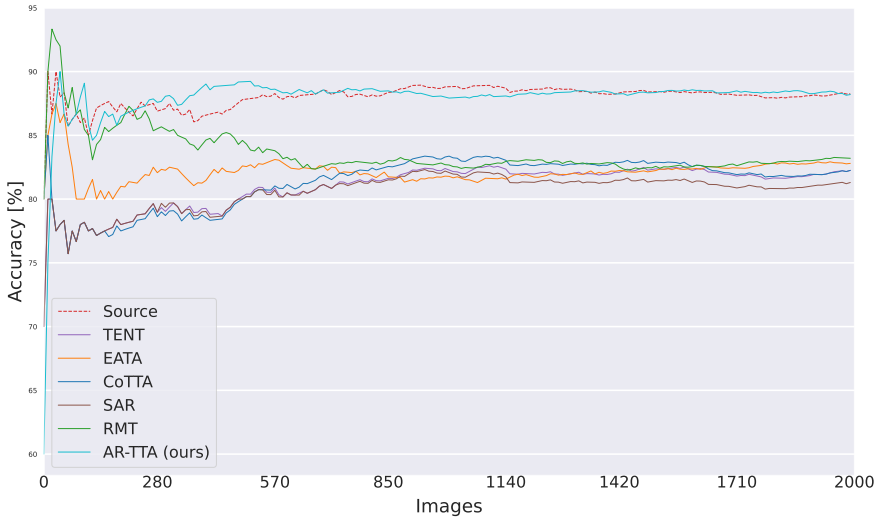


Figure 5: Batch-wise classification accuracy (%) averaged in a window of 400 batches on CIFAR10.1 benchmark for the chosen methods continually adapted to the sequences of data. Best viewed in color.

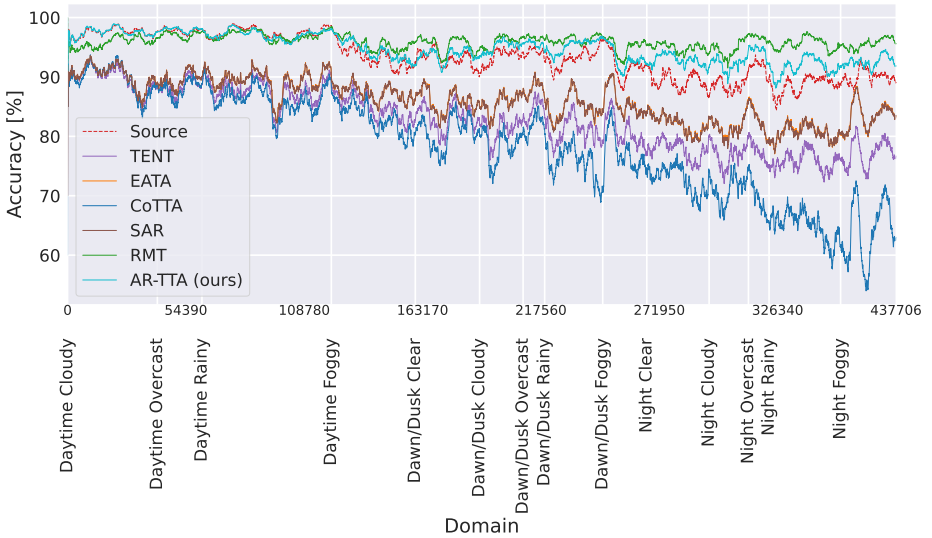


Figure 6: Batch-wise classification accuracy (%) averaged in a window of 500 batches on SHIFT-C benchmark for the chosen methods continually adapted to the sequences of data. The major ticks on the x-axis symbolize the beginning of the next sequence and, at the same time, a different domain. Minor ticks on the x-axis (numbers) indicate the image number. Best viewed in color.

Table 4: Classification accuracy (%) for the standard CIFAR10C online continual test-time adaptation task.

Method														Mean
	Gaussian	shot	impulse	defocus	glass	motion	zoom	snow	frost	fog	brightness	contrast	elastic-trans	
Source	27.7	34.3	27.1	53.1	45.7	65.2	58.0	74.9	58.7	74.0	90.7	53.3	73.4	56.5
BN-1	67.3	69.4	59.7	82.7	60.4	81.4	83.0	78.1	77.7	80.6	87.3	83.4	71.4	75.0
TENT [15]	67.9	71.4	62.5	83.2	62.9	82.1	83.8	79.5	79.7	81.4	87.8	84.3	73.5	76.7
EATA [8]	70.3	74.9	67.1	83.0	65.6	82.3	84.0	80.3	81.4	82.2	88.0	85.1	74.7	78.2
CoTTA [16]	72.5	76.4	70.5	80.6	66.6	78.3	80.1	75.8	77.0	77.1	83.8	77.3	72.0	75.7
SAR [9]	67.4	69.6	60.8	82.6	61.4	81.5	82.8	78.1	77.7	80.5	87.4	83.4	71.5	75.2
RMT w/o replay [9]	73.2	77.5	72.1	79.6	69.8	78.2	79.7	76.9	78.0	79.9	83.0	81.6	76.5	77.6
RMT [9]	74.8	78.6	74.8	85.5	73.1	84.7	86.1	84.2	85.4	87.2	90.1	89.1	82.2	83.1
AR-TTA (Ours) w/o replay	69.5	73.6	63.3	83.5	63.0	82.5	84.5	80.2	80.4	81.9	88.4	83.8	74.2	77.3 \pm 0.07
AR-TTA (Ours)	69.2	74.8	66.4	84.5	67.8	83.7	85.2	81.4	82.7	83.4	88.0	84.7	73.9	78.8 \pm 0.13

3 Implementation Details

The results are averaged between 3 random seeds. Samples from CIFAR10C and ImageNet-C are shuffled. Considering the sequential nature of data in CLAD-C and SHIFT-C benchmarks (video sequences), we did not want to shuffle images. Instead, we trained 3 source models with 3 different seeds and averaged the results between experiments with different models. We test the method in a continual manner on every benchmark, which means that the methods continually adapt the models without the reset to the source state in between the domains.

Implementations of the compared methods were taken from their official code repositories. We use all hyper-parameters and optimizers suggested by the papers or found in the code. We follow the standard model architectures used in TTA experiments and use WideResnet28 for CIFAR10C and CIFAR10.1, and ResNet50 for ImageNet-C, CLAD-C, and SHIFT-C. Moreover, since we use a smaller batch size (BS) of 10 and benchmarks that have not been used before in TTA, we search for the optimal learning rate (LR) for each method. We focus on lowering the LR, considering the decreased batch size. Additionally, we search for the ϵ hyperparameter of EATA to correctly reject samples for adaptation. The results of the parameter search can be found in Table 5. The details and parameters used for each method are described below.

TENT [15] We use Adam optimizer with LR = 0.00025 for CIFAR10.1 and LR = 0.00003125 for every other tested dataset. In the original paper, TENT uses LR = 0.001 for all the datasets except ImageNet, but it performed worse with this value in our setup.

CoTTA [16] Adam optimizer with LR = 0.00025 is used for every tested benchmark, except ImageNet-C for which LR was equal to 0.00003125. The original implementation set LR to 0.001, but with an adjusted value, it achieved better results. We follow the suggestions for other hyperparameter values given by the authors. The restoration probability p is set to 0.01, the smoothing factor of the exponential moving average of teacher weights α is equal to 0.999, and the confidence threshold for applying augmentations p_{th} is set to 0.92.

EATA [8] We use the SGD optimizer with a momentum of 0.9 and LR of 0.00025 for CIFAR10C, ImageNet-C, and CLAD-C. LR for SHIFT-C is equal to 0.00003125 and 0.001 for CIFAR10.1. The original EATA paper uses an LR value of 0.005/0.00025 for

CIFAR10C/ImageNet-C, but they used $BS = 64$. After the search for the optimal ε parameter value for filtering redundant samples, we set it to 0.05/0.6 for CLAD-C/SHIFT. The value of ε for CIFAR10C and CIFAR10.1/ImageNet-C is equal to 0.4/0.05, as in the original paper. The entropy constant E_0 is set to the standard value of $0.4 \times \ln C$, where C was the number of classes, following the original paper and [24]. The trade-off parameter β is equal to 1, and 2000 samples are used to calculate the fisher importance of model weights as for the CIFAR10 dataset in the original paper.

SAR [24] SGD optimizer is used with the momentum of 0.9 and $LR = 0.001$ for both CIFAR10C and CIFAR10.1, and $LR = 0.00025$ for ImageNet-C, CLAD-C, and SHIFT-C. It almost aligns with the authors' choice since, in original experiments, they used a learning rate equal to 0.00025/0.001 for ResNet/Vit models. The parameter E_0 is set to $0.4 \times \ln C$, as in the paper, similarly to EATA. We follow the authors' choice of a constant reset threshold value e_0 of 0.2, and a moving average factor equal to 0.9. The radius parameter ρ is set to the default value of 0.05.

RMT [24] Adam optimizer is used with $LR = 0.00025$ for CIFAR10C, CLAD-C, and SHIFT-C. LR is equal to 0.00003125 for CIFAR10.1 and ImageNet-C, all following the grid search. Following the original implementation, we use temperature τ for contrastive loss set to 0.1 and momentum param α utilized to update the mean teacher equal to 0.999.

AR-TTA (Ours) We use an SGD optimizer with momentum of 0.9. We set LR of 0.00025 for both ImageNet-C and CIFAR10.1, and 0.001 for the rest of the benchmarks. The scale hyper-parameter γ is set to 0.1 for CIFAR10.1, CLAD-C, and SHIFT-C. It is equal to 10 for ImageNet-C and CIFAR10C. α value for weighting the β_{ema} is equal to 0.2. We set the initial β_{ema} value to 0.1. The ψ and ρ parameters used for beta distribution to sample λ for mixup is equal to the standard value of 0.4. We store 2000 of exemplars from source data for memory replay.

4 Benchmarks

4.1 CIFAR10C And ImageNet-C Benchmarks Details

CIFAR10C and ImageNet-C are widely used datasets in TTA. They involve training the source model on train split of clean CIFAR10/ImageNet datasets [24, 25] and test-time adaptation on CIFAR10C/ImageNet-C. CIFAR10C and ImageNet-C consist of images from clean datasets which were modified by 15 types of corruptions with 5 levels of severity [26]. They were first used for evaluating the robustness of neural network models and are now widely utilized for testing the adaptation capabilities of TTA methods. We test the adaptation on a standard sequence of the highest corruption severity level 5, frequently utilized by previous approaches [24, 25, 26]. For ImageNet-C we utilize a subset of 5000 samples for each corruption, based on *RobustBench* library [27], following [26].

4.2 CIFAR10.1 Benchmark Details

CIFAR10.1 [28] was designed to minimize the distribution shift relative to the original CIFAR10 dataset [29]. It contains roughly 2000 test images. The images in CIFAR10.1 are

Table 5: Mean classification accuracy (%) for CIFAR10C, ImageNet-C, CIFAR10.1, CLAD-C, and SHIFT-C continual test-time adaptation task for compared state-of-the-art methods with different learning rates and EATA’s ϵ parameter.

Method	learning rate	ϵ	CIFAR10C	CIFAR10.1	CLAD-C	SHIFT-C	ImageNet-C
CoTTA [14]	0.001	-	49.3	79.3	71.5	74.3	3.8
	0.00025	-	75.7	82.3	71.8	78.6	10.6
	0.00003125	-	74.5	81.8	71.8	76.2	15.3
TENT [15]	0.001	-	24.3	81.2	64.4	63.4	0.6
	0.00025	-	72.3	82.3	71.0	75.3	3.1
	0.00003125	-	76.7	81.4	71.1	82.7	29.3
SAR [9]	0.001	-	75.2	81.3	70.6	86.0	11.3
	0.00025	-	75.1	81.3	70.6	86.0	31.5
	0.00003125	-	75.0	81.3	70.6	86.0	28.8
RMT [9]	0.001	-	83.0	81.1	76.0	93.1	30.2
	0.00025	-	83.1	81.9	75.3	95.9	28.6
	0.00003125	-	81.5	83.3	74.5	93.1	30.5
EATA [9]	0.001	0.60	-	82.6	70.1	80.4	-
	0.001	0.40	76.3	82.9	70.6	80.4	-
	0.001	0.10	-	82.4	70.6	86.0	-
	0.001	0.05	-	82.4	70.6	86.0	27.3
	0.00025	0.60	-	82.4	70.5	85.6	-
	0.00025	0.40	78.2	82.6	70.6	86.1	-
	0.00025	0.10	-	82.4	70.6	86.0	-
	0.00025	0.05	-	82.4	70.7	86.0	31.7
	0.00003125	0.60	-	82.4	70.6	86.1	-
	0.00003125	0.40	76.5	82.4	70.6	86.0	-
	0.00003125	0.10	-	82.4	70.6	86.0	-
	0.00003125	0.05	-	82.4	70.6	86.0	31.6

a subset of the TinyImages dataset [16]. The source model utilized for testing on this benchmark was pre-trained on the original CIFAR10 dataset.

4.3 CLAD-C Benchmark Details

CLAD-C [14] is an online classification benchmark for autonomous driving with the goal of introducing a more realistic testing bed for continual learning. It consists of natural, temporal correlated, and continuous distribution shifts created by utilizing the data from SODA10M dataset [5]. The images taken at different locations, times of day, and weather, are chronologically ordered, inducing distribution shifts in labels and domains.

The classification task was created by cutting out the annotated 2D bounding boxes of six classes and using them as separate images for classification. Bounding boxes with fewer than 1024 pixels were discarded. The images are padded by their shortest axis (modify the aspect ratio to 1:1) and resized to 32x32. For the ResNet50 model, we additionally resize them to 224x224.

Since it is designed for testing the continual learning setup and the model is originally supposed to be trained sequentially on the train sequences, we slightly modify the setup and pre-train the source model on the first train sequence. TTA is continually tested on the 5 remaining ones with a total number of 17092 images.

4.4 SHIFT-C Benchmark Details

The SHIFT-C benchmark is created using the SHIFT dataset [17]. It consists of multiple types of autonomous driving data from the CARLA Simulator [9]. We used RGB images from the front view of a car with discrete domain shifts and bounding box annotations. More

specifically, we download the required data with the script from SHIFT’s website <https://www.vis.xyz/shift/>, using the following command:

```
python download.py --view "front" \
--group "[img, det_2d]" \
--split "[train, val]" \
--framerate "images" \
--shift "discrete" TARGET_DIR
```

To load the data for experiments, we utilized *shift-dev* repository: <https://github.com/SysCV/shift-dev>.

We create an image classification task data, following the steps from the CLAD-C[14] benchmark. Bounding boxes in the dataset are categorized into six classes, and so are the created images. Example images are displayed in Figure 7. We present a class distribution in Figure 8.

We distinguish between domains by the course annotations of time of day and weather. The source model is trained on images from train split, taken in the daytime in clear weather. The TTA is also tested on data from the train split but from different weather conditions and times of the day. Details about the size of each domain can be found in Table 6.



Figure 7: Example images sourced from various domains within the SHIFT-C benchmark.

Table 6: The number of samples in each domain in SHIFT-C benchmark

Domain nr	Time of day	Weather	Number of images
Source data		clean	57039
1	daytime	cloudy	41253
2		overcast	20497
3		rainy	59457
4		foggy	38590
5		clear	29543
6	dawn/dusk	cloudy	19985
7		overcast	9901
8		rainy	26677
9		foggy	20258
10		clear	28639
11	night	cloudy	18068
12		overcast	9471
13		rainy	32864
14		foggy	25464
Sum			437706

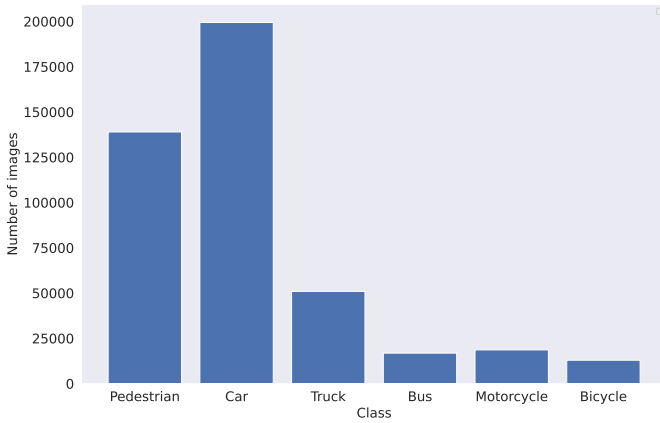


Figure 8: SHIFT-C benchmark class distribution.

References

- [1] Francesco Croce, Maksym Andriushchenko, Vikash Sehwal, Edoardo DeBenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.
- [2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [3] Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. *CoRR*, abs/2211.13081, 2022.
- [4] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [5] Jianhua Han, Xiwen Liang, Hang Xu, Kai Chen, Lanqing Hong, Jiageng Mao, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Xiaodan Liang, and Chunjing Xu. Soda10m: A large-scale 2d self/semi-supervised object detection dataset for autonomous driving, 2021.
- [6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.
- [7] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.
- [8] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Minghui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore*,

- Maryland, USA, volume 162 of *Proceedings of Machine Learning Research*, pages 16888–16905. PMLR, 2022.
- [9] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofu Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.
- [10] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018. <https://arxiv.org/abs/1806.00451>.
- [11] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020.
- [12] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21371–21382, June 2022.
- [13] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008. doi: 10.1109/TPAMI.2008.128.
- [14] Eli Verwimp, Kuo Yang, Sarah Parisot, Lanqing Hong, Steven McDonagh, Eduardo Pérez-Pellitero, Matthias De Lange, and Tinne Tuytelaars. Clad: A realistic continual learning benchmark for autonomous driving. *Neural Networks*, 161:659–669, 2023.
- [15] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [16] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 7191–7201. IEEE, 2022.