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

Damian Sójka[1,2], Sebastian Cygert[2,3], Bartłomiej Twardowski[2,4,5], and Tomasz Trzciński[2,6,7]
[1]Poznań University of Technology, [2]IDEAS NCBR, [3]Gdańsk University of Technology,
[4]Autonomous University of Barcelona, [5]Computer Vision Center, [6]Warsaw University of Technology, [7]Tooploox

## Abstract

*Test-time adaptation is a promising research direction that allows the source model to adapt itself to changes in data distribution without any supervision. Yet, current methods are usually evaluated on benchmarks that are only a simplification of real-world scenarios. Hence, we propose to validate test-time adaptation methods using the recently introduced datasets for autonomous driving, namely CLAD-C and SHIFT. We observe that current test-time adaptation methods struggle to effectively handle varying degrees of domain shift, often resulting in degraded performance that falls below that of the source model. We noticed that the root of the problem lies in the inability to preserve the knowledge of the source model and adapt to dynamically changing, temporally correlated data streams. Therefore, we enhance well-established self-training framework by incorporating a small memory buffer to increase model stability and at the same time perform dynamic adaptation based on the intensity of domain shift. The proposed method, named AR-TTA, outperforms existing approaches on both synthetic and more real-world benchmarks and shows robustness across a variety of TTA scenarios.*

## 1. Introduction

Test-time adaptation (TTA) aims to adapt the source data pretrained model to the current data distribution on-the-fly during test-time, using an unlabeled stream data [16, 14]. Those methods are required to work well in a wide range of challenging setups, including temporal correlation between consecutive frames and lengthy sequences with gradual and abrupt domain shifts.

Existing approaches are based on self-training methods such as using pseudo-labels or prediction entropy minimization [17, 16]. However, when tested over lengthy sequences with changing distributions, those methods can become unstable, and as a result, the self-training feedback becomes noisier and performance degrade [1]. Moreover, without using any source data, the model is prone to *catastrophic forgetting* [9] of initially acquired knowledge.

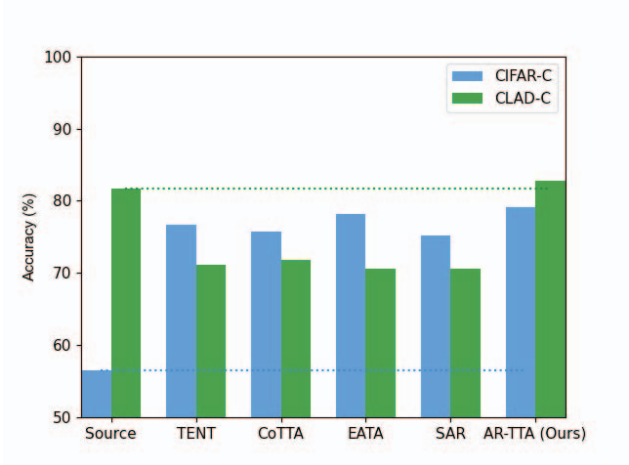


Figure 1. Continual test-time adaptation methods evaluated on synthetic (CIFAR-10C) and realistic (CLAD-C) domain shifts. Our method is the only one that consistently allows to improve over the naive strategy of using the (frozen) source model.

TTA approaches are mostly evaluated on datasets with synthetically generated domain shifts (*e.g.* image corruptions [5]) or on relatively short-length sequences [16, 17, 2] and as such it is not known how those methods will work in real-life scenarios with unlimited streams of data. Therefore, we adapt the autonomous driving benchmark for continual learning CLAD-C [15] to the continual test-time adaptation setting. Moreover, we use a realistic, synthetically generated driving dataset SHIFT [13] to create a SHIFT-C benchmark.

In the proposed evaluation setup, we find out that current approaches lack the required stability, as their performance significantly deteriorates compared to the source model, see Figure 1. Additionally, we notice that they struggle to correctly estimate batch norm (BN) statistics with temporally correlated data streams and low batch sizes. We propose a method in which we extend a popular self-training framework [17] with a small memory buffer, which is used during adaptation to prevent knowledge forgetting, without rely-

ing on heuristic-based strategies or resetting model weights that are often used [17, 11]. Thanks to using mixup data augmentation [18], a relatively small number of samples are required. Further, we develop a module for dynamic batch norm statistics adaptation, which interpolates the calculated statistics between those of the pretrained model and those obtained during deployment, based on the intensity of domain shift. We call our method **AR-TTA**, as we improve **A**daptation by using dynamic batch norm statistics and maintain knowledge by **R**epeating samples from the memory buffer combined with mixup data augmentation.

## 2. Related Work

**Test-time adaptation (TTA).** TTA setup is a type of domain adaptation in which there is no access to the source data and the model adapts to the test-time distribution on-the-fly in an *online* fashion based on unlabeled test data. TENT [16] uses prediction entropy minimization to update only batch-norm weights. EATA [10] improves efficiency and reliability by using diverse samples with low prediction entropy and incorporating EWC regularization. CoTTA [17] updates the entire model using techniques like weight averaging and stochastic model restoration, which randomly resets the model weights to the source model state. SAR [11] removes noisy samples and adds loss components for flat minima but still employs model reset to prevent forgetting.

**TTA benchmarks.** Test-time adaptation benchmarking typically involves using synthetic corruptions proposed in [5] individually, allowing model reset between domains. However, in practical applications, the target distribution can continually change over time. For this reason, continual test-time adaptation, introduced in [17], eliminates model resets at domain boundaries. Yet, the distribution shifts arising in the real world may be very different from the synthetic ones. Therefore, recently a CLAD autonomous driving benchmark [15] was created. It consists of real-world images with naturally occurring distribution shifts like changes in weather and lighting conditions, traffic intensity, etc. In this work, we use it for test-time adaptation, that is without using any label information.

## 3. Method

The overview of our method is presented in Figure 2.

**Weight-averaged Consistency.** Following previous works [4, 17], we propose to employ self-training on pseudo-labels and keep two models, where one is updated by the exponential moving average of another's weights. We initialize two identical artificial neural network models, student model $f_\theta$ and teacher model $f_{\theta'}$, with the same weights obtained by training on source data. For each batch of test data $\mathbf{x}_t^T$ at time step $t$ we generate predictions from

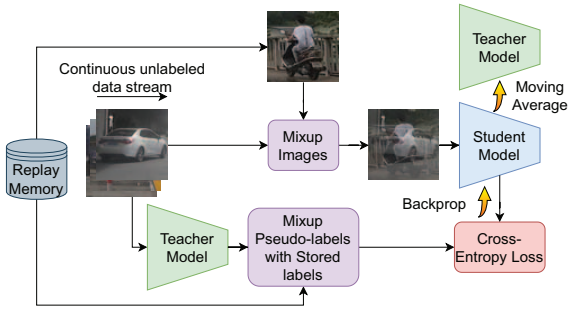


Figure 2. Our AR-TTA method adapts to unlabeled, continual data streams using saved exemplars from source pretraining. It involves two twin neural network models: teacher and student. Exemplars are sampled from memory and augmented with test images by mixup augmentation. Pseudo-labels from the teacher model are mixed up with labels from memory. The student model is updated using cross-entropy loss between its predictions and augmented pseudo-labels. The teacher model is adapted using an exponential moving average of the student's weights. Predictions come from the teacher model.

both models. Teacher model predictions $\hat{y'}_t^T$ are used as soft pseudo-labels. The student model is updated by the cross-entropy loss between its predictions and the pseudo-labels:

$$\mathcal{L}_{\theta_t}(\mathbf{x}_t^T) = -\sum_c \hat{y'}_{t,c}^T \log \hat{y}_{t,c}^T \tag{1}$$

where $\hat{y}_{t,c}^T$ is the probability of class $c$ predicted by the student model. Lastly, the teacher's weights $\theta'$ are updated by the exponential moving average of the student's weights $\theta$.

In contrast to most of the existing approaches [10, 16, 11], we adapt every weight of a model. The final predictions for the current test batch $\mathbf{x}_t^T$ are the classes with the highest probabilities in pseudo-labels generated by the teacher model before the update.

**Experience Replay with Adaptation.** To alleviate the issue of catastrophic forgetting and strengthen the model's initial knowledge, we use the class-balanced replay buffer combined with the mixup data augmentation [18], inspired by a few of the continual learning works [7, 19].

We save a predefined number of random exemplars from labeled source data in the memory. In each test-time adaptation iteration, we randomly sample exemplars $\mathbf{x}_t^S$, along with their labels $\mathbf{y}_t^S$, from memory. The number of sampled exemplars is equal to the batch size. Mixupped batch of samples $\tilde{\mathbf{x}}_t$ and pseudo-labels $\tilde{\mathbf{y}}_t$ is generated by linearly interpolating test data with data from memory:

$$\tilde{\mathbf{x}}_t = \lambda \mathbf{x}_t^T + (1-\lambda)\mathbf{x}_t^S \quad \text{and} \quad \tilde{\mathbf{y}}_t = \lambda \hat{\mathbf{y}'}_t^T + (1-\lambda)\mathbf{y}_t^S \tag{2}$$

where $\lambda \sim \text{Beta}(\psi, \psi)$, for $\psi \in (0, \infty)$, and $\hat{\mathbf{y}}_t'^T$ is a matrix of pseudo-labels produced by the teacher model based on the current unmodified test batch. Student model takes

Table 1. Classification accuracy (%) for the standard CIFAR10-to-CIFAR10C online continual test-time adaptation task.

$t \longrightarrow$

| Method | *Gaussian* | *shot* | *impulse* | *defocus* | *glass* | *motion* | *zoom* | *snow* | *frost* | *fog* | *brightness* | *contrast* | *elastic_trans* | *pixelate* | *jpeg* | Mean Acc. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source | 67.57 | 68.92 | 49.86 | 83.61 | 61.67 | 76.41 | 80.2 | 76.81 | 75.82 | 70 | 84.65 | 65.16 | 75.32 | 69.91 | **80.36** | 56.5 |
| BN stats adapt | 67.3 | 69.1 | 59.6 | 82.7 | 60.3 | 81.5 | 83.1 | 78.0 | 78.0 | 80.3 | 87.2 | 83.2 | 71.2 | 75.2 | 68.0 | 75.0 |
| TENT-continual [16] | 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 | 78.2 | 71.6 | 76.7 |
| EATA [10] | 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 | **80.1** | 73.8 | 78.2 |
| CoTTA [17] | **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 | 75.5 | 72.2 | 75.7 |
| SAR [11] | 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 | 68.2 | 75.2 |
| Ours (AR-TTA) | 68.4 | 73.7 | 66.2 | **84.5** | 66.2 | **83.6** | **85.8** | **81.4** | **82.8** | **84.1** | **89.5** | **88.1** | **77.6** | 78.0 | 76.4 | **79.1** |

Table 2. Classification accuracy (%) for the CLAD-C continual test-time adaptation task.

$t \longrightarrow$

| Method | T1 | T2 | T3 | T4 | T5 | Mean Day Acc. | Mean Night Acc. | Avg. Mean Class Acc. | Min Mean Class Acc. | Mean Acc. |
|---|---|---|---|---|---|---|---|---|---|---|
| Source | 76.4 | **86.0** | 75.5 | 86.5 | 68.5 | 86.2 | 73.1 | **54.1** | **4.8** | 81.7 |
| BN stats adapt | 72.2 | 69.0 | 74.6 | 74.3 | 62.2 | 71.1 | 69.6 | 41.7 | 1.6 | 70.6 |
| TENT-continual [16] | 72.4 | 68.9 | 76.3 | 75.3 | 61.9 | 71.4 | 70.3 | 40.1 | 0.0 | 71.1 |
| EATA [10] | 72.2 | 69.1 | 74.6 | 74.3 | 62.1 | 71.2 | 69.6 | 41.8 | 1.6 | 70.6 |
| CoTTA [17] | 74.8 | 67.6 | **79.2** | 76.2 | 65.1 | 71.1 | 73.2 | 38.6 | 0.0 | 71.8 |
| SAR [11] | 72.1 | 69.1 | 74.6 | 74.3 | 62.2 | 71.2 | 69.6 | 41.7 | 1.6 | 70.6 |
| Ours (AR-TTA) | **78.0** | 85.9 | 78.8 | **87.1** | **69.6** | **86.4** | **75.3** | **54.0** | **4.8** | **82.6** |

augmented batch $\tilde{\mathbf{x}}_t$ as input. Its predictions are compared with interpolated labels $\tilde{\mathbf{y}}_t$ to calculate the loss.

**Dynamic Batch Normalization Statistics.** Due to domain shift, state-of-the-art test-time adaptation methods [17, 10, 16, 11] usually discard statistics calculated during training and estimate data distribution based on each batch of data separately. However, this way of calculating the statistics is flawed since the sample size from data is usually too small to correctly estimate the data distribution, depending on the batch size.

Therefore, we take the inspiration from [6] and propose to use BN statistics from source data to estimate BN statistics $\phi_t = (\mu_t, \sigma_t)$ at time step $t$ during test-time by linearly interpolating between saved statistics from source data $\phi^S$ and calculated values from current batch $\phi_t^T$:

$$\phi_t = (1 - \beta)\phi^S + \beta\phi_t^T \tag{3}$$

where $\beta$ is a parameter that weights the influence of saved and currently calculated statistics.

Since the severity of distribution shift might vary, we utilize the symmetric KL divergence as a measure of distance between distributions $D(\phi_{t-1}, \phi_t^T)$ to adjust the value of $\beta$ accordingly:

$$D(\phi_{t-1}, \phi_t^T) = \frac{1}{C}\sum_{i=1}^{C} KL(\phi_{t-1,i}||\phi_{t,i}^T) + KL(\phi_{t,i}^T||\phi_{t-1,i}) \tag{4}$$

$\beta_t$ at time step $t$ is calculated as follows:

$$\beta_t = 1 - e^{-\gamma D(\phi_{t-1}, \phi_t^T)} \tag{5}$$

where $\gamma$ is a scale hyperparameter.

To provide more stability for the adaptation, we take into account previous $\beta_{1:t-1}$ values and use an exponential moving average for $\beta_t$ update:

$$\beta = (1 - \alpha)\beta_{t-1} + \alpha\beta_t \tag{6}$$

where $\alpha$ is a hyperparameter.

## 4. Experiments

**Baselines.** To evaluate the performance of our method and validate its efficacy in handling realistic domain shifts, we conduct experiments involving five state-of-the-art methods as baselines in image classification task: TENT-continual [16], EATA [10], CoTTA [17], and SAR [11]. Moreover, we show results for discarding BN statistics from source data and calculating the statistics for each batch separately (BN stats adapt) [12]. Additionally, we showcase the results obtained from the frozen source model to verify the effectiveness of adaptation (Source).

**Artificial Domain Shifts.** For comparing the TTA methods on standard artificial domain shifts, we utilize CIFAR10-to-CIFAR10C task [16, 17], which includes sequentially adapting to different image corruption types on the 5th level of severity. Following other state-of-the-art TTA methods, we use the pretrained WideResnet28 model from *RobustBench* [3] model zoo. The results are shown in Table 1. Artificial domain shifts pose a great challenge for the source model, achieving only 56.5% mean accuracy. Calculating BN statistics for each batch separately already significantly improves the result to 75% accuracy on corrupted images. Each of the compared state-of-the-art TTA methods uses the BN stats adapt technique. Therefore their performance improves over it, but the increase in accuracy value is not sig-

Table 3. Classification accuracy (%) for the SHIFT-C continual test-time adaptation task.

| Method | $t \longrightarrow$ daytime | | | | dawn/dusk | | | | | night | | | | | Avg. Mean Class Acc. | Min Mean Class Acc. | Mean Acc. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *cloudy* | *overcast* | *rainy* | *foggy* | *clear* | *cloudy* | *overcast* | *rainy* | *foggy* | *clear* | *cloudy* | *overcast* | *rainy* | *foggy* | | | |
| Source | **97.6** | **97.9** | **97.4** | 93.1 | **92.8** | **93.8** | **93.8** | **93.4** | 91.7 | 88.8 | 89.2 | **91.2** | 88.6 | 90.0 | 91.3 | 86.8 | 93.4 |
| BN stats adapt | 90.2 | 89.9 | 89.2 | 87.2 | 85.9 | 85.7 | 88.2 | 84.7 | 85.5 | 81.9 | 82.0 | 81.1 | 80.5 | 84.0 | 71.8 | 58.8 | 86.0 |
| TENT-continual [16] | 89.6 | 88.8 | 87.5 | 84.6 | 83.3 | 81.2 | 85.0 | 80.7 | 80.2 | 78.0 | 77.0 | 76.1 | 75.7 | 77.6 | 57.6 | 35.4 | 82.7 |
| EATA [10] | 90.2 | 90.0 | 89.3 | 87.3 | 86.0 | 86.0 | 88.2 | 84.9 | 85.8 | 82.0 | 82.2 | 81.3 | 80.7 | 84.1 | 72.6 | 60.8 | 86.1 |
| CoTTA [17] | 89.7 | 88.2 | 86.1 | 82.7 | 80.9 | 78.3 | 82.2 | 75.3 | 76.7 | 72.9 | 71.2 | 67.4 | 64.3 | 64.9 | 50.6 | 19.6 | 78.6 |
| SAR [11] | 90.2 | 89.9 | 89.2 | 87.2 | 85.9 | 85.7 | 88.2 | 84.7 | 85.5 | 81.9 | 82.0 | 81.1 | 80.5 | 84.0 | 71.8 | 58.8 | 86.0 |
| Ours (AR-TTA) | 97.2 | 97.5 | 96.8 | **95.2** | 91.6 | 93.5 | 93.5 | 92.5 | **93.9** | **89.9** | **91.5** | **91.2** | **89.2** | **92.5** | **91.5** | **89.0** | **93.7** |

nificant. Our method AR-TTA outperforms all of the compared techniques achieving 79.0% of mean accuracy. This shows the effectiveness of our method on the standard continual TTA test benchmark.

**Natural Domain Shifts.** Our experiments on natural domain shifts involve utilizing CLAD-C and SHIFT-C benchmarks. Since CLAD-C 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 pretrain the source model on the first train sequence. TTA is continually tested on the 5 remaining ones.

The SHIFT-C benchmark is created from the SHIFT dataset. The source model is trained on images taken in clear weather during the day, and the adaptation methods are tested in various weather and time of day combinations.

We utilize ResNet50 architecture with weights pretrained on ImageNet obtained from *torchvision* library [8] and finetuned to the source data for the specific benchmark.

Results for CLAD-C are shown in Table 2. Calculating BN statistics for each batch (BN stats adapt) does not improve the performance over the frozen source model and degrades the mean accuracy. Similarly, the state-of-the-art TTA methods achieve significantly lower results compared to the frozen source model, rendering them not effective for natural domain shifts. It suggests that benchmarking such methods on artificial domain shifts in the form of corruptions is not a valuable estimate of the TTA method's performance in practical applications. Moreover, it shows that keeping the precalculated statistics intact might sometimes be more beneficial for less severe domain shifts, on which the source model performs relatively well. Our method, which uses precalculated statistics and exemplars of source data during adaptation, outperformed state-of-the-art methods and achieves higher accuracy than the source model, which shows the effectiveness and adapting capabilities.

Similar conclusions can be drawn from SHIFT-C benchmark results in Table 3. The frozen source model achieves impressive results, while state-of-the-art methods significantly degrade it. Moreover, the adaptation schemes of TENT and CoTTA methods caused the accuracy to be lower than the simple BN stats adaptation approach. Only AR-TTA was able to improve the source model performance during TTA.

Table 4. Mean classification accuracy (%) for CIFAR10C and CLAD-C tasks for different configurations of the proposed method.

| Method | CIFAR10C | CLAD-C |
|---|---|---|
| **A** Pseudo-labels | 75.4 | 70.6 |
| **B** + Weight-avg. teacher | 75.6 | 70.6 |
| **C** + Dynamic BN stats | 76.0 | 82.2 |
| **D** + Replay memory | 78.3 | 82.2 |
| **E** + Mixup | **79.1** | **82.6** |

## 5. Conclusion

We evaluate existing continual test-time adaptation (TTA) methods in real-life scenarios using more realistic data. Our findings reveal that current state-of-the-art methods are inadequate in such settings, as they fall short in achieving accuracy comparable to the frozen source model. To address these limitations, we propose a novel and straightforward method called AR-TTA, based on the self-training framework. AR-TTA utilizes a small memory buffer of source data, combined with mixup data augmentation, and dynamically updates the batch norm statistics based on the intensity of domain shift.

Through experimental studies, we demonstrate that the AR-TTA method achieves state-of-the-art performance on various benchmarks. Notably, AR-TTA consistently outperforms the source model, which serves as the ultimate baseline for feasible TTA methods. Our more realistic evaluation of TTA with a variety of different datasets provides a better understanding of their potential benefits and shortcomings.

## Acknowledgements

Tomasz Trzcinski is supported by National Centre of Science (Poland) Grant No. 2022/45/B/ST6/02817. Tomasz Trzciński is also supported by National Centre of Science (Poland) Grant No. 2020/39/B/ST6/01511.

## References


[1] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 627–636. Computer Vision Foundation / IEEE, 2019.

[2] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 295–305. IEEE, 2022.

[3] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[4] Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. *CoRR*, abs/2211.13081, 2022.

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[6] Junyuan Hong, Lingjuan Lyu, Jiayu Zhou, and Michael Spranger. Mecta: Memory-economic continual test-time adaptation. In *ICLR*, 2023.

[7] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2022.

[8] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. `https://github.com/pytorch/vision`, 2016.

[9] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[10] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui 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.

[11] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *Internetional Conference on Learning Representations*, 2023.

[12] 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.

[13] 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.

[14] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 9229–9248. PMLR, 2020.

[15] 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.

[16] 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.

[17] 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.

[18] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[19] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14306–14318. Curran Associates, Inc., 2021.