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, Wroclaw, Poland

[6] Gdańsk University of Technology
Gdańsk, Poland

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 the 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. The code is available at https://github.com/dmn-sjk/AR-TTA.

1 Introduction

Deep neural networks have been shown to achieve remarkable performance in various tasks, however, they perform very well only when the test-time distribution is close to the training-time distribution. This poses a significant challenge since in a real-world application a domain shift can occur in many circumstances, *e.g.*, weather change, time of day shift, or sensor

degradation. For this reason, Test-Time Adaptation (TTA) methods have been widely developed in recent years [29, 31]. They aim to adapt the source data pre-trained model to the current data distribution on-the-fly during test-time, using an unlabeled stream of test data.

An effective TTA method should work well regardless of a wide range of challenges, encompassing both abrupt and gradual domain shifts. Moreover, the model should be able to maintain stable performance even when handling lengthy sequences, potentially extending indefinitely. Existing approaches are based on updating model parameters using pseudo-labels or entropy regularization [31, 33]. Further, filtering the less reliable samples is often employed to reduce error accumulation and improve the computational efficiency [1, 21, 22]. However, all of the methods can become unstable due to the aforementioned difficulties, accumulate errors, make the pseudo-labels noisier, and cause performance degradation [4]. Without using any source data the model is prone to *catastrophic forgetting* [19] of previously acquired knowledge.

Currently, methods are mostly evaluated on datasets with artificial, unrealistic domain shifts or relatively short-length sequences [5, 31, 33], hence it is unknown how those methods will work in real-life scenarios with natural domain shifts. Therefore, we adapt the autonomous driving benchmark CLAD [30], to the continual adaptation setting. Moreover, we use a SHIFT dataset [28], which is synthetically generated by utilizing the realistic autonomous driving simulator CARLA [9], provides very long sequences, and allows us to specifically control for different domain factors (time of day, weather conditions).

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 (Fig. 1). Additionally, we notice that they struggle to correctly estimate batch norm statistics with temporally correlated data streams and low batch sizes.
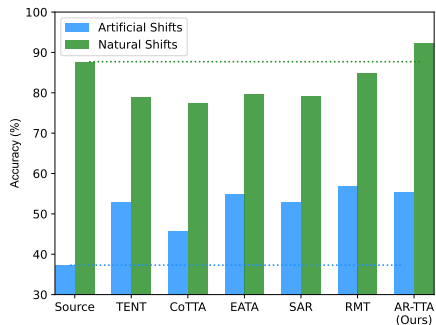

Figure 1: Continual test-time adaptation methods evaluated on artificial (CIFAR10C, ImageNet-C [10]) and natural (CIFAR10.1 [26], SHIFT [28], CLAD-C [30]) domain shifts. Our method is the only one that consistently allows to improve over the naive strategy of using the (frozen) source model.

In our TTA method for an image classification task, we extend popular self-training framework [33] with a small memory buffer, which is used during adaptation to prevent knowledge forgetting, without relying on heuristic-based strategies or resetting model weights that are often used [22, 33]. Thanks to using mixup data augmentation [34], a relatively small number of samples is required. Furthermore, we develop a module for dynamic batch norm statistics adaptation, which interpolates the calculated statistics of the pre-trained model and those obtained during testing, 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. As a result, our proposed method AR-TTA is simple, stable, and works well across a range of datasets with different shift intensities, when using small batches of data and over very long sequences. Our main contributions can be summarized as follows:

- We propose novel evaluation benchmarks for TTA based on autonomous-driving scenarios, which show significant limitations of existing TTA methods.

- Based on discovered vulnerabilities, we propose a TTA method that dynamically updates the batch norm statistics based on the intensity of domain shift and use small memory buffer combined with mixup data augmentation.

- Extensive evaluation shows that the proposed method obtains state-of-the-art performance on benchmarks with both artificial distortions and natural domain shifts.

2 Related Work

Test-time adaptation (TTA). Domain adaptation methods can be split into different categories based on what information is assumed to be available during adaptation [32]. While in some scenarios, access to some labels in target distribution is available, the most common scenario is *unsupervised domain adaptation*, in which the model has access to labeled source data and unlabeled target data at adaptation time. Additionally, in *test-time adaptation* the model needs to adapt to the test-time distribution on-the-fly, in an *online* fashion. In the test-time training (TTT) method [29] the model solves self-supervised tasks on the incoming batches of data to update its parameters. TENT [31] updates only batch-norm weights to minimize predictions entropy. EATA [21] further improves the efficiency of test-time adaptation methods, by using only diverse and reliable samples (with low prediction entropy). Additionally, it uses EWC [13] regularization to prevent drastic changes in parameters important for the source domain. Contrary to the TENT and EATA approaches, CoTTA [33] updates the whole model. To prevent performance degradation it uses exponential weight averaging as well as stochastic model restoration, where randomly selected weights are reset to the source model. SAR [22] further improves by removing noisy test samples with large gradients and using sharpness-aware minimization. Nevertheless, they also use model reset, to prevent forgetting. RMT [8] propose a mean teach framework and adapt it based on contrastive loss, self-training loss, and a simple replay strategy utilizing all of the source training data during test-time. Moreover, it assumes an additional warm-up step before the adaptation. Our method uses a small number of replay exemplars and does not require any extra steps before TTA.

TTA benchmarks. The most popular setting for test-time adaptation includes applying different classes of synthetic corruptions proposed in [10] to commonly used datasets, like CIFAR10 [14] or ImageNet [7] to artificially obtain domain shifts. Another popular dataset for continual test-time adaptation is DomainNet [23] which consists of images in different domains (e.g., sketches, infographics). Yet, the distribution shifts arising in real-world deployment, e.g., in autonomous driving, may be very dissimilar from different renditions of the same objects. Hence, recently a CLAD, autonomous driving benchmark [30] was introduced. It consists of naturally occurring distribution shifts like changes in weather and lighting conditions, traffic intensity, etc. It was developed for the supervised continual learning scenario. In this work, we use it for test-time adaptation, that is without using any label information. In our work, we also use SHIFT benchmark [28], a synthetically generated dataset for autonomous driving that captures the continuously evolving nature of the real world. Similarly to us, CoTTA [33] includes realistic domain shifts but their test is very small (1600 images). To sum up, we extend over previous TTA work by focusing on realistic continual domain shifts over very long sequences.
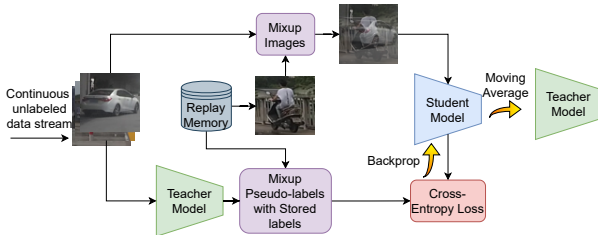
Figure 2: Our method, AR-TTA, utilizes a replay strategy and the mean teacher framework. Each image is paired with an exemplar sampled from memory and image pairs are mixed up. Similarly, pseudo-labels from the teacher model are mixed up with the labels of sampled exemplars. The student model is updated based on cross-entropy loss between its predictions on augmented samples and augmented pseudo-labels. The teacher model is adapted based on an exponential moving average of student's weights. Predictions for each image are taken from the teacher model.

3   Method

Continual TTA aims to adapt the pre-trained model f_{θ_0} trained on the labeled source data $(\mathcal{X}^S, \mathcal{Y}^S)$ to the ever-changing stream of unlabeled test data batches $\mathbf{x}^T$ on-the-fly during the evaluation. Our proposed approach (Fig. 2.) can be divided into three parts. We start the description by introducing the model update procedure in Section 3.1. Then, in Section 3.2 we explain the usage of experience replay with mixup augmentation. The process of adapting batch normalization statistics is presented in Section 3.3

3.1   Weight-averaged Consistency

Following previous works [8, 33], we use the self-training mean teacher framework to update the model. It improves the reliability of pseudo-labels and minimizes the risk of error accumulation. We initialize two identical artificial neural network models, student model f_θ and teacher model $f_{\theta'}$, with the equal weights obtained by training on source data. For each batch of test data $\mathbf{x}_t^T$ at time step t, predictions are derived from 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:

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

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

$$\theta_{t+1}' = \alpha \theta_t' + (1-\alpha)\theta_{t+1} \qquad (2)$$

where α is a smoothing factor. Using a weight-averaged teacher model ensures less noisy pseudo-labels [33], improves generatlization [20], and added inertia prevents rapid weights update based on noisy self-training feedback.

We do not limit the weights update only to affine parameters of batch normalization layers, as in many other TTA methods [21, 22, 51], and we adapt the whole model. We argue that adapting only batch normalization layer weights does not give the model enough

flexibility to perform successfully on varying domains. We confirm this claim experimentally in the appendix.

3.2 Experience Replay with Adaptation

During continual test-time adaptation, self-training feedback is not guaranteed to be accurate, and frequent model updates inevitably strive for significant error accumulation, catastrophic forgetting, and even model collapse [22]. Continual learning works show that experience replay is one of the most effective strategies for mitigating catastrophic forgetting [2, 3, 24, 30]. Following this insight we propose to use the replay strategy during adaptation to remind the model what it has learned and strengthen its initial knowledge. Building on this, we integrate Mixup data augmentation [34] to enhance model robustness, inspired by works such as [15, 35].

After completing the pre-training of the source model, we store a low, equal for each class, number of random exemplars from the 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. Mixed up batch of samples $\tilde{\mathbf{x}}_t$ is generated by linearly interpolating samples from test data with samples from memory:

$$\tilde{\mathbf{x}}_t = \lambda \mathbf{x}_t^T + (1 - \lambda)\mathbf{x}_t^S \tag{3}$$

where $\lambda \sim \text{Beta}(\psi, \rho)$, for $\psi, \rho \in (0, \infty)$. Similarly, labels for cross-entropy loss are the result of interpolation between pseudo-labels produced by the teacher model based on the current unmodified test batch $\hat{\mathbf{y}}_t^{\prime T}$ and labels $\mathbf{y}_t^S$ from the memory, with the same λ:

$$\tilde{\mathbf{y}}_t = \lambda \hat{\mathbf{y}}_t^{\prime T} + (1 - \lambda)\mathbf{y}_t^S \tag{4}$$

Student model takes augmented batch $\tilde{\mathbf{x}}_t$ as input. Its predictions are compared with interpolated labels $\tilde{\mathbf{y}}_t$ to calculate the loss as described in the previous Section 3.1.

A similar approach to mixing exemplars from replay memory with the ones to train on was successfully used in LUMP [15], however, they used this method for the continual representation learning task. Using experience replay along with the Mixup augmentation helps the model preserve already obtained knowledge and makes the noisiness of pseudo-labels less impactful for the adaptation process.

3.3 Dynamic Batch Norm Statistics

Batch normalization [12] (BN) layers normalize network values using running statistics from initial training data, applied during test-time. However, out-of-distribution data can disrupt this process, leading to poor model performance. Test-time adaptation methods [21, 22, 31, 33] often recalculate statistics per each batch separately. However, this approach can be flawed due to small sample sizes and temporal data correlation. In these cases, using BN statistics from source data may offer a more accurate distribution estimation.

To perform a robust statistics estimation, we take the inspiration from [11] and propose to calculate BN statistics $\phi_t = (\mu_t, \sigma_t)$ at time step t during test-time by linearly interpolating between saved statistics from source data ϕ^S and calculated values from current batch ϕ_t^T:

$$\phi_t = (1 - \beta_{ema})\phi^S + \beta_{ema}\phi_t^T \tag{5}$$

where β_{ema} is a parameter that weights the influence of saved and currently calculated statistics. Since the severity of the distribution shift might vary, we need to adequately adjust the value of β_{ema}. It should be large in cases when the distribution shift is severe compared to the source domain and low when the distributions are similar. Following [11], we utilize the symmetric KL divergence as a measure of distance between distributions $D(\phi_{t-1}, \phi_t^T)$:

$$D(\phi_{t-1}, \phi_t^T) = \frac{1}{C} \sum_{i=1}^{C} KL(\phi_{t-1,i} || \phi_{t,i}^T) + KL(\phi_{t,i}^T || \phi_{t-1,i}) \qquad (6)$$

The distance is used to calculate β_t at time step t:

$$\beta_t = 1 - e^{-\gamma D(\phi_{t-1}, \phi_t^T)} \qquad (7)$$

where γ is a scale hyperparameter. To compensate for the fact that the current distribution can be wrongly estimated and to provide more stability for the adaptation, we take into account previous $\beta_{1:t-1}$ values and use an exponential moving average for β_{ema} using α coefficient:

$$\beta_{ema} = (1 - \alpha)\beta_{t-1} + \alpha\beta_t \qquad (8)$$

Our method differs from MECTA [11] by preserving the original BN statistics instead of using exponential moving averages. We are motivated by the fact that changing this value causes the inevitable forgetting, and susceptibility to temporal correlation. We ensure consistent performance on similar domains to the source data. At the same time, we allow for a slight drift away from source data statistics, by using the exponential moving average of β parameter, giving enough flexibility for the adaptation to severe domain shifts.

4   Experiments

Datasets and Benchmarks. We evaluate the methods on multiple image classification datasets with domain shifts. We use CIFAR10C and ImageNet-C as benchmarks with artificial, corruption-based shifts, widely utilized for evaluation of the test-time adaptation approaches [8, 21, 22, 31, 33]. The experiments with natural shifts include tests on CIFAR10.1 [25] and SHIFT [28] datasets, and CLAD-C continual learning benchmark [30] adapted to the test-time adaptation setting.

CIFAR10.1 was designed to minimize the distribution shift relative to the original CIFAR10 dataset. This allows us to test the robustness of TTA methods to very delicate distribution shifts. The goal of CLAD-C benchmark was to introduce a more realistic testing bed for continual learning. The images taken at different locations and conditions are chronologically ordered, inducing distribution shifts in labels and domains. It consists of 6 test sequences. We 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. In the SHIFT dataset, images are taken in numerous types of realistic domains simulated in a virtual environment, including different weather and times of day. We consider it a natural domain shift benchmark due to its realistic representation of distribution shifts, which are induced by real conditions rather than image corruptions. We train the source model on images taken in clear weather during the day and test the adaptation methods for various weather combinations and times of day. We end up with 14 different domains resulting in a total of 380667 images. We called our original continual test-time adaptation benchmark the SHIFT-C. More details can be found in the appendix.

Methodology. We examine the continual test-time adaptation setup where the model is continually adapted to new domains without resetting the model unless it is part of a tested method. To simulate the continuous stream of data and the need for the model to adapt quickly, we use a low batch size of 10. This is also the batch size commonly used in the online continual learning [16]. In embedded applications, limited resources often mean smaller batch sizes, necessitating TTA methods to work effectively with fewer samples. We assess the methods using mean classification accuracy and average mean class accuracy (AMCA). The latter calculates the mean accuracy over all classes averaged for each domain, making every class equally important in class-imbalanced datasets.

Baselines. We conduct experiments involving six state-of-the-art methods as baselines: TENT-continual [31], EATA [21], CoTTA [33], SAR [22], and RMT [8]. Moreover, we show results for discarding BN statistics from source data and calculating the statistics for each batch separately (BN-1) [27]. We also report the performance of the frozen source model (Source). To provide a fair comparison, the results without a replay strategy of RMT and our method (AR-TTA) are presented.

Implementation Details. Following other state-of-the-art TTA works, we use pre-trained WideResnet28/ResNet50 models from *RobustBench* [6] model zoo for CIFAR10C and CIFAR10.1/ImageNet-C datasets. On the rest of the benchmarks, we utilize ResNet50 architecture with weights pre-trained on ImageNet obtained from *torchvision* library [17] and finetuned to the source data for the specific benchmark.

Our method utilizes the default replay memory size of 2000 samples, which is commonly used in continual learning settings [18] and adds only minor storage requirements. The RMT method utilizes all of the source training data, following the original work. The rest of the details regarding the technical side of experiments can be found in the appendix.

4.1 Results

The results for all of the tested benchmarks are presented in Table 1. Our method can achieve the best average accuracy of 75.5% indicating the robustness of our approach. Moreover, it shows a solid performance even without the replay method.

Artificial Domain Shifts. Artificial domain shifts pose a great challenge for source models, achieving only 56.5%/17.1% mean accuracy for CIFAR10C/ImageNet-C. Calculating BN statistics for each batch separately, already significantly improves the result to 75.0%/26.9% accuracy on corrupted images. Each of the compared state-of-the-art TTA methods uses the BN-1 technique, therefore their performance improves over it, but the increase in accuracy value is not that significant. RMT outperformed our method on CIFAR10C, however, it lacks performance on other benchmarks and utilizes the whole source training dataset for replay, rendering AR-TTA a more reliable method for different kinds of datasets.

Natural Domain Shifts. The BN-1 method significantly degrades the performance of the frozen source model on natural shifts by about 8-10 percentage points of accuracy. Similarly, the state-of-the-art TTA methods achieve significantly lower mean accuracy than the Source, rendering them ineffective for natural domain shifts. It suggests that benchmarking such methods on artificial domain shifts in the form of corruptions is not a reliable estimate of the TTA method's performance in practical applications. Our method, which uses pre-calculated statistics and exemplars of source data during adaptation, outperformed state-of-the-art methods and achieved higher accuracy than the source model. This indicates the effectiveness and adapting capabilities. Keeping the pre-calculated statistics intact might

Table 1: Classification accuracy (%) for all of the tested online continual test-time adaptation tasks. Methods that use exemplars are in the right section. Red color indicates accuracy lower compared to a source model.

Method	CIFAR10C	ImageNet-C	CIFAR10.1	CLAD-C	SHIFT-C	Average
Source	56.5	18.1	88.3	81.3	93.5	67.5
BN-1	75.0	26.9	81.3	71.1	85.1	67.9
TENT [🔲]	76.7	29.2	82.3	71.5	82.7	68.5
EATA [🔲]	78.2	31.5	82.9	71.1	85.1	69.8
CoTTA [🔲]	75.7	15.5	82.3	72.6	77.4	64.7
SAR [🔲]	75.2	30.8	81.3	71.1	85.1	68.7
RMT w/o replay [🔲]	77.6	21.7	80.6	75.1	92.2	69.4
RMT [🔲]	**83.1**	30.5	83.3	75.3	**95.9**	73.6
AR-TTA (Ours) w/o replay	$77.3_{\pm0.07}$	$30.0_{\pm0.45}$	$88.2_{\pm0.10}$	$\mathbf{83.9}_{\pm0.30}$	$92.4_{\pm0.25}$	74.4
AR-TTA (Ours)	$\mathbf{78.8}_{\pm0.13}$	$\mathbf{32.0}_{\pm0.07}$	$\mathbf{88.3}_{\pm0.05}$	$83.7_{\pm0.64}$	$94.8_{\pm0.03}$	**75.5**

sometimes be more beneficial, especially on domain shifts on which the source model performs relatively well.

Accuracy Over Time. Results in Figure 3 show how the accuracy of state-of-the-art methods drops below the source model performance. We can see that AR-TTA maintains the accuracy of the source model on the first two domains where the source model works well, and then improves over it when the source model performance deteriorates (at domain T3).

Table 2: Classification accuracy and average mean class accuracy (AMCA) (%) for the CLAD-C continual test-time adaptation task.

Method	T1	T2	T3	T4	T5	Mean day	Mean night	Mean	AMCA
Source	75.6	85.9	73.3	87.5	66.2	86.6	71.2	81.3	57.6
BN-1	73.2	69.9	75.0	75.5	59.7	72.2	69.1	71.1	48.3
TENT [🔲]	73.4	69.8	76.5	76.1	59.7	72.4	69.8	71.5	47.6
EATA [🔲]	73.3	69.9	75.0	75.6	59.7	72.2	69.1	71.1	48.4
CoTTA [🔲]	75.2	69.3	80.2	77.0	62.7	72.4	72.9	72.6	44.8
SAR [🔲]	73.2	69.9	75.0	75.5	59.7	72.2	69.1	71.1	48.3
RMT w/o replay [🔲]	**87.1**	70.9	**86.6**	76.9	64.3	73.9	**79.3**	75.1	48.4
RMT [🔲]	83.8	71.3	85.0	77.6	66.4	74.5	78.4	75.3	48.8
AR-TTA (Ours) w/o replay	76.9	**86.7**	81.4	87.9	**73.5**	87.2	77.1	$\mathbf{83.9}_{\pm0.30}$	$59.6_{\pm2.92}$
AR-TTA (Ours)	77.2	**86.7**	80.0	**89.6**	70.7	**87.8**	75.7	$83.7_{\pm0.64}$	$\mathbf{63.1}_{\pm3.32}$

Table 3: Classification accuracy and average mean class accuracy (AMCA) (%) for the SHIFT-C continual test-time adaptation task.

Method	daytime				dawn/dusk					night					Mean	AMCA
	cloudy	overcast	rainy	foggy	clear	cloudy	overcast	rainy	foggy	clear	cloudy	overcast	rainy	foggy		
Source	**97.9**	**98.2**	**97.5**	92.5	93.6	94.1	94.0	93.5	91.5	89.1	89.3	90.6	89.1	90.7	93.5	89.5
BN-1	89.1	88.9	88.0	86.2	85.3	84.8	87.3	83.5	84.8	81.3	81.2	80.3	79.6	83.5	85.1	69.9
TENT [🔲]	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	82.7	57.6
EATA [🔲]	89.1	88.9	88.0	86.2	85.3	84.8	87.4	83.6	84.9	81.4	81.4	80.3	79.7	83.7	85.1	70.5
CoTTA [🔲]	88.2	87.1	84.1	80.5	78.7	76.2	80.5	74.0	74.9	71.5	70.3	67.3	64.9	66.2	77.4	47.2
SAR [🔲]	89.1	88.9	88.0	86.2	85.3	84.8	87.3	83.5	84.8	81.3	81.2	80.3	79.6	83.6	85.1	69.9
RMT w/o replay [🔲]	93.0	94.2	93.6	91.7	91.0	91.3	93.0	92.2	90.8	90.7	90.9	91.7	90.5	91.7	92.0	80.3
RMT [🔲]	95.9	97.2	97.0	**95.4**	95.4	95.9	96.5	96.3	95.0	94.9	95.1	96.3	95.5	95.6	95.9	91.1
AR-TTA (Ours) w/o replay	96.4	96.5	95.3	93.2	92.2	91.9	93.2	91.4	91.8	88.7	88.7	88.6	87.5	91.2	$92.4_{\pm0.25}$	$83.5_{\pm0.96}$
AR-TTA (Ours)	97.7	98.0	97.4	94.3	94.2	95.5	94.8	95.2	93.1	92.3	92.7	93.0	91.4	92.6	$94.8_{\pm0.03}$	$90.2_{\pm0.24}$

Table 4: Classification accuracy (%) for CI-FAR10C and CLAD-C tasks for different configurations of the proposed method.

Method	CIFAR10C	CLAD-C
A: Weight-avg. teacher	$75.7_{\pm 0.07}$	$71.1_{\pm 0.53}$
B: A + Replay memory	$77.3_{\pm 0.16}$	$69.0_{\pm 0.66}$
C: B + Mixup	$78.5_{\pm 0.13}$	$72.2_{\pm 0.31}$
D: A + Dynamic BN stats	$77.3_{\pm 0.07}$	$83.8_{\pm 0.82}$
E: D + Replay memory	$79.8_{\pm 0.03}$	$82.8_{\pm 1.09}$
AR-TTA (Ours): E + Mixup	$78.8_{\pm 0.13}$	$83.7_{\pm 0.64}$

Table 5: Classification accuracy and average mean class accuracy (AMCA) (%) results for state-of-the-art methods with simple **replay method added**.

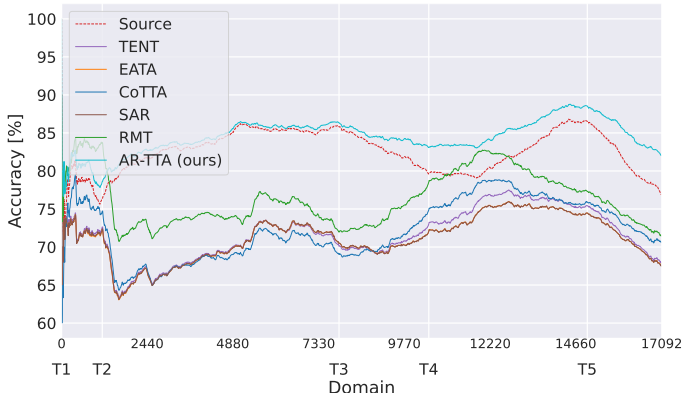| Method	CIFAR10C	CLAD-C	
	Mean	Mean	AMCA
Source	56.5	81.3	57.6
TENT [☐]	77.3	70.3	49.2
EATA [☐]	78.6	71.1	48.4
CoTTA [☐]	**79.9**	72.6	51.0
SAR [☐]	75.3	71.1	48.3
AR-TTA (Ours)	78.8	**83.7**	**63.1**

Figure 3: Batch-wise classification accuracy (%) averaged in a window of 400 batches on CLAD-C benchmark for the chosen methods continually adapted to the sequence of data, with major ticks on the x-axis symbolizing the beginning of a different domain and minor ticks indicating image number. Best viewed in color.

4.2 Detailed Analysis

Effect Of Exemplars. Tables 2 and 3 present the AMCA results for CLAD-C and SHIFT-C benchmarks. The results show a significant improvement in mean per-class accuracy while using a replay method. This suggests that the usage of replay memory might be important for keeping the performance for each class high in class-imbalanced TTA setups. Also, on average, our method is still better than that of our competitors when all methods have used exemplars and is the only one that consistently improves over the frozen model.

Component Analysis. Table 4 shows the contribution of individual components used in the proposed method. For the initial setup **A**, we used a weight-averaged teacher model to generate pseudo-labels and cross-entropy loss to adapt the model of which the BN statistics from source data are discarded. Adding a simple replay method (**B**, **E**) by injecting randomly augmented exemplars from memory to the batch in a 1:1 ratio, did not improve the performance on every dataset. It can be seen that mixup data augmentation can boost the performance of a simple replay method (**C**). Moreover, dynamic BN statistics significantly contribute to the accuracy increase (**D**, **E**, **AR-TTA**), especially on the CLAD-C benchmark.

Baselines With Simple Replay Memory. To ensure a fair comparison and check the state-of-the-art approaches' performance with the use of a replay strategy, we added a simple replay method to each of them. A constant number of replay exemplars, equal to the batch

size, was sampled on each batch from a class-balanced replay memory. The value of cross entropy loss calculated from exemplars was added to the original loss of each method. The results are in the Table 5. While the proposed method performs slightly worse than CoTTA on the CIFAR10C, it performs significantly better on the natural domain shift dataset. Most importantly, our method is the only one that constantly improves over the source model.

Computational Efficiency Evaluation Table 6 presents a comparison of wall-clock time and memory usage for baseline methods and our approach on 10,000 images from CIFAR10C benchmark. While our method does not rank as the most computationally efficient, it achieves a balance between computational demands and performance. Notably, despite incorporating exemplars, we maintain a consistent computational budget. This is because the additional samples are utilized within the mix up augmentation, leaving the training batch size unchanged. This approach contrasts with the RMT method, where the use of replay techniques substantially increases computational complexity.

Table 6: The wall-clock time (seconds) and memory usage (MB) measured for processing 10,000 images of CIFAR10C on a single RTX 4080 GPU.

Method	Time [s]	Memory [MB]
Source	8.0	304
BN-1	8.3	304
TENT [■]	16.3	506
EATA [□]	24.3	505
CoTTA [■]	319.4	1532
SAR [□]	30.8	506
RMT [■] w/o replay	55.5	1576
RMT [■]	163.7	3039
AR-TTA (Ours) w/o replay	66.2	1098
AR-TTA (Ours)	66.6	1098

5 Conclusion

In this paper, we evaluate existing continual test-time adaptation (TTA) methods in real-life scenarios using more realistic data by proposing two new evaluation benchmarks, namely SHIFT-C and CLAD-C. Our findings reveal that current state-of-the-art methods are inadequate in such settings, as they fall short of achieving accuracies better than the frozen source model. This raises concerns about the applicability of certain TTA methods in the real world and sheds light on the frequent model resets observed in some approaches. 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. These benchmarks include realistic evaluations with small batch sizes, long test sequences, varying levels of domain shift, as well as artificial scenarios such as corrupted CIFAR10-C. 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.

Limitations. The main limitation of our method is that we use a memory buffer from the source data, which might be an issue in resource-constrained scenarios or if there are some privacy concerns.

Acknowledgments

References

[1] Motasem Alfarra, Hani Itani, Alejandro Pardo, Shyma Alhuwaider, Merey Ramazanova, Juan C Pérez, Zhipeng Cai, Matthias Müller, and Bernard Ghanem. Revisiting test time adaptation under online evaluation. *arXiv preprint arXiv:2304.04795*, 2023.

[2] M. Boschini, L. Bonicelli, P. Buzzega, A. Porrello, and S. Calderara. Class-incremental continual learning into the extended der-verse. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(05):5497–5512, may 2023. ISSN 1939-3539.

[3] Arslan Chaudhry, Albert Gordo, Puneet Kumar Dokania, Philip H. S. Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *AAAI Conference on Artificial Intelligence*, 2019.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

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

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448–456. JMLR.org, 2015.

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[15] 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.

[16] Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomput.*, 469(C):28–51, jan 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.10.021. URL https://doi.org/10.1016/j.neucom.2021.10.021.

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

[18] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[19] 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.

[20] Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[21] 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.

[22] 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.

[23] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019.

[24] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5533–5542, 2017.

[25] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10?, 06 2018.

[26] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? 2018. https://arxiv.org/abs/1806.00451.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[33] 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.

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

[35] 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.