Robust Mean Teacher for Continual and Gradual Test-Time Adaptation

Mario Döbler *  Robert A. Marsden *  Bin Yang
University of Stuttgart
{mario.doebler, robert.marsden, bin.yang}@iss.uni-stuttgart.de

Abstract

Since experiencing domain shifts during test-time is inevitable in practice, test-time adaption (TTA) continues to adapt the model after deployment. Recently, the area of continual and gradual test-time adaptation (TTA) emerged. In contrast to standard TTA, continual TTA considers not only a single domain shift, but a sequence of shifts. Gradual TTA further exploits the property that some shifts evolve gradually over time. Since in both settings long test sequences are present, error accumulation needs to be addressed for methods relying on self-training. In this work, we propose and show that in the setting of TTA, the symmetric cross-entropy is better suited as a consistency loss for mean teachers compared to the commonly used cross-entropy. This is justified by our analysis with respect to the (symmetric) cross-entropy’s gradient properties. To pull the test feature space closer to the source domain, where the pre-trained model is well posed, contrastive learning is leveraged. Since applications differ in their requirements, we address several settings, including having source data available and the more challenging source-free setting. We demonstrate the effectiveness of our proposed method “robust mean teacher” (RMT) on the continual and gradual corruption benchmarks CIFAR10C, CIFAR100C, and Imagenet-C. We further consider ImageNet-R and propose a new continual DomainNet-126 benchmark. State-of-the-art results are achieved on all benchmarks. 1

1. Introduction

Assuming that training and test data originate from the same distribution, deep neural networks achieve remarkable performance. In the real world, this assumption is often violated for a deployed model, as many environments are non-stationary. Since the occurrence of a data shift [35] during test-time will likely result in a performance drop, domain generalization aims to improve robustness and generalization already during training [11, 13, 32, 43, 45]. However, these approaches are often limited, due to the wide range of potential data shifts [30] that are unknown during training. To gain insight into the current distribution shift, recent approaches leverage the test samples encountered during model deployment to adapt the pre-trained model. This is also known as test-time adaptation (TTA) and can be done either offline or online. While offline TTA assumes to have access to all test data at once, online TTA considers the setting where the predictions are needed immediately and the model is adapted on the fly using only the current test batch.

While adapting the batch normalization statistics during test-time can already significantly improve the performance [38], more sophisticated methods update the model weights using self-training based approaches, like entropy minimization [49]. However, the effectiveness of most TTA methods is only demonstrated for a single domain shift at a time. Since encountering just one domain shift is very unlikely in real world applications, [50] introduced continual test-time adaptation where the model is adapted to a sequence of domain shifts. As pointed out by [50], adapting the model to long test sequences in non-stationary environments is very challenging, as self-training based methods are prone to error accumulation due to miscalibrated predictions. Although it is always possible to reset the model after it has been updated, this prevents exploiting previously acquired knowledge, which is undesirable for the following reason: While some domain shifts occur abruptly in practice, there are also several shifts which evolve gradually over time [17]. In [26], this setting is denoted as gradual test-time adaptation. [17, 26] further showed that in the setting of gradual shifts, pseudo-labels are more reliable, resulting in a better model adaptation to large domain gaps. However, if the model is reset and the domain gap increases over time, model adaptation through self-training or self-supervised learning may not be successful [17, 24].

To tackle the aforementioned challenges, we introduce a robust mean teacher (RMT) that exploits a symmetric cross-entropy (SCE) loss instead of the commonly used cross-entropy (CE) loss to perform self-training. This is motivated by our findings that the CE loss has undesirable gradient

---

*Equal contribution.

1Code is available at: https://github.com/mariodoebler/test-time-adaptation
properties in a mean teacher framework which are compensated for when using an SCE loss. Furthermore, RMT uses a multi-viewed contrastive loss to pull test features towards the initial source space and learn invariances with regards to the input space. While our framework performs well for both continual and gradual domain shifts, we observe that mean teachers are especially well suited for easy-to-hard problems. We empirically demonstrate this not only for gradually shifting test sequences, but also for the case where the domain difficulty with respect to the error of the initial source model increases. Since source data might not be available during test-time due to privacy or accessibility reasons, recent approaches in TTA focus on the source-free setting. Lacking labeled source data, source-free approaches can be susceptible to error accumulation. Therefore, as an extension to our framework, we additionally look into the setting where source data is accessible.

We summarize our contributions as follows:

- By analyzing the gradient properties, we motivate and propose that in the setting of TTA, the symmetric cross-entropy is better suited for a mean teacher than the commonly used cross-entropy.

- We present a framework for both continual and gradual TTA that achieves state-of-the-art results on the existing corruption benchmarks, ImageNet-R, and a new proposed continual DomainNet-126 benchmark.

- For our framework, we address a wide range of practical requirements, including the source-free setting and having source data available.

## 2. Related Work

### Domain Generalization

Generalizing to unseen test distributions is the area which is generally addressed by domain generalization. It has been shown that data augmentation during training improves generalizability and robustness. Another direction is to learn domain-invariant features, as addressed by. The idea of domain randomization uses different synthesis parameters of simulation environments during the learning process to improve generalization.

### Unsupervised Domain Adaptation (UDA)

Since generalizing to every unseen target domain remains an open question, unsupervised domain adaptation considers the setting, where next to labeled source data, unlabeled target data is also available. This makes the problem easier since the available target data narrows down the domain shift. To increase the performance on the target domain, one line of work focuses on minimizing the discrepancy between the source and target feature distributions, by using either adversarial learning, discrepancy based loss functions, or contrastive learning.

While it is also possible to adapt the domains in the input space, self-training has emerged as a powerful technique. It uses the network’s predictions as pseudo-labels to minimize the (cross-)entropy for target data. To increase the reliability of the network’s predictions, rely on a mean teacher.

### Test-time Adaptation (TTA)

Test-time adaptation methods adapt a pre-trained model after deployment leveraging the currently available test data. Since the test samples also provide insights into the distribution shift, showed that simply adapting the batch normalization (BN) statistics during test-time can already significantly improve the performance on corrupted data. This is in spirit to which previously proposed to update the BN statistics in the setting of UDA. While this strategy only requires a forward pass, current approaches in TTA further perform a backward pass in which the model weights are updated. For example, minimizes the entropy with respect to the BN parameters. minimizes the entropy with respect to all parameters and uses test-time augmentation to artificially increase the batch size. Other methods apply contrastive learning or even introduce an additional self-supervision loss during source pre-training that is later leveraged to perform the adaptation during test-time. Diversity regularizers are a recent approach to circumvent collapse to trivial solutions potentially caused by confidence maximization. While many methods assume to have a batch of test data available, one line of work focuses on single-sample TTA.

### Continual and Gradual Test-time Adaptation

Test-time adaptation methods typically adapt the model to a single target domain. In the real world, encountering a sequence of domain shifts is much more likely. Therefore, continual test-time adaptation considers the case of online TTA with sequentially changing domains. Some of the existing TTA methods can also be applied in the continual setting, such as the online version of TENT, Diversity regularizers are a recent approach to circumvent collapse to trivial solutions potentially caused by confidence maximization. This is a result of miscalibrated predictions, as highlighted in. CoTTA was the first method specifically proposed for the setting of continual TTA. It uses weight and augmentation-averaged predictions to address the problem of error accumulation, which are further combined with a stochastic restore to circumvent catastrophic forgetting. proposes a method that not only considers a continual setting, but also investigates scenarios where shifts are gradual. Based on the theory of that self-training is sufficient as long as the encountered shifts are small enough, GTTA introduces intermediate domains by mixup or style transfer for effective self-training.
This technique, known as self-training, can only work if re-labels to update itself has been shown to be very effective. Adaptation by using the network’s predictions as pseudo-labels provide only inadequate predictions for the current test data and pull test features towards the initial source space, and optionally source replay depending on whether source data is available. We empirically show that performing a mean teacher warm-up is further beneficial. An overview of our framework is depicted in Fig. 1.

3. Methodology

In many practical applications, the environmental conditions change over time. Hence, a model $f_{th}$, pre-trained on source data $(X^S, Y^S)$, can quickly become sub-optimal and provide only inadequate predictions for the current test data $X^T_t$ at time step $t$. To prevent the model’s performance from deterioration, online test-time adaptation updates the model based on the current test data $x^T_t$. Depending on the application, labeled source data $(X^S, Y^S)$ may be available.

In this work, we propose a robust mean teacher that leverages the symmetric cross-entropy, which we show to have more desirable gradient properties, contrastive learning to become invariant to small changes in the input space and pull test features towards the initial source space, and optionally source replay depending on whether source data is accessible. We empirically show that performing a mean teacher warm-up is further beneficial. An overview of our framework is depicted in Fig. 1.

3.1. Robust Mean Teacher

Conducting unsupervised domain adaptation or test-time adaptation by using the network’s predictions as pseudo-labels to update itself has been shown to be very effective. This technique, known as self-training, can only work if reliable pseudo-labels are provided. One possibility to improve the pseudo-labels are mean teachers (MT) [42]. By simply averaging the weights of a student model over time, the resulting teacher model provides a more accurate prediction function than the final function of the student [42].

To realize the MT framework, a student and a teacher model are initialized at time step $t = 0$ with source pre-trained weights $\theta_0$. During test-time, the student $f_{th}$ is updated ($\theta_t \rightarrow \theta_{t+1}$) by minimizing the cross-entropy (CE)

$$L_{CE}(q, p) = -\sum_{c=1}^{C} q_c \log p_c$$

between the softmax predictions of the student $p$ and the teacher $q$, with $C$ being the total number of classes. Subsequently, the non-trainable weights of the teacher $\theta_t^{\theta}$ are updated using an exponential moving average $\theta_{t+1}^{\theta} = \alpha \theta_t^{\theta} + (1 - \alpha) \theta_{t+1}$, where $\alpha$ denotes the momentum of how much the teacher is updated. Due to the exponential moving average, mean teachers are also known to be more stable in changing environments [18], which is also a desirable property for continual and gradual test-time adaptation.

Undesirable gradient properties Looking at the binary case of the cross-entropy, the partial derivative with respect to the student’s output probability $p$ is given by

$$\frac{\partial L_{CE}}{\partial p} = \frac{p - q}{p - p^2},$$

where $p$ and $q$ are the student’s and teacher’s output probabilities, respectively. First, when both student and teacher have the same confidence $p = q \in [0.5, 1.0]$, the resulting gradient is zero. So even if student and teacher agree with the same confidence that a sample belongs to the same class $p = q > 0.5$, there will be no progress. Suppose we have a small shift where our classifier is still able to make correct predictions, albeit with less certainty. If we did not update the decision boundary, another small shift could lead to incorrect predictions, impairing self-training. Second, the gradient is highly imbalanced, especially for the case when either the teacher or student is very confident. The gradient explodes when the student is very confident in contrast to the teacher, resulting in the student to become a lot less confident. On the contrary, when the teacher is very confident, but the student is not, learning is limited due to the relatively small gradient. These effects are illustrated in the form of the loss surface in Fig. 2 and the gradient surface in Fig. 3 in the appendix.

Symmetric cross-entropy has better gradient properties

Next, we propose the usage of the symmetric cross-entropy (SCE) [51] for a mean teacher and hypothesize that in this setting it has better gradient properties compared to the

![Diagram of RMT framework](image)
commonly used cross-entropy loss. The symmetric cross-entropy was originally proposed as a more robust version of the cross-entropy with regards to label-noise [51], which is also desirable in our context. For two distributions \( q \) and \( p \), the symmetric cross-entropy is defined as

\[
L_{SCE}(q, p) = - \sum_{c=1}^{C} q_c \log p_c - \sum_{c=1}^{C} p_c \log q_c, \quad (3)
\]

where the first term corresponds to the cross-entropy loss \( L_{CE} \), while the second term is the reverse cross-entropy (RCE) loss \( L_{RCE} \).

For the binary SCE loss, the gradient is given by

\[
\frac{\partial L_{SCE}}{\partial p} = \frac{\partial L_{CE}}{\partial p} + \log(1/q - 1). \quad (4)
\]

In the case where the teacher has a probability of \( q = 0.5 \), the derivative of the SCE loss is equivalent to the derivative of the cross-entropy loss. Looking again at the situation where both student and teacher have the same confidence \( p = q \in (0.5, 1.0) \), the absolute value of the derivative is larger the more confident the teacher. This leads to increasing the student’s confidence, potentially benefiting the self-training process. Moreover, the additional term of the SCE derivative results in a more balanced gradient due to the second term originating from the RCE derivative: \( \log(1/q - 1) < 0 \) for \( q > 0.5 \).

Looking at the properties of the cross-entropy loss from another perspective, it is already known that for the CE loss, samples with low confidences dominate the overall gradient [33]. This can be obstructive in the setting of self-training, since low confidence samples are typically more inaccurate. Following the analysis of [5], the partial derivative of the RCE loss \( L_{RCE} \) with respect to the \( j \)-th output element of the network is given by

\[
\frac{\partial L_{RCE}}{\partial z_j} = p_j \left( \sum_{c=1}^{C} p_c \log q_c - \log q_j \right). \quad (5)
\]

Now, if the probability vector \( p \) is kept fixed, it becomes apparent that the gradient reaches its maximum when \( q \) is a one-hot vector, while the minimum is obtained for a uniform probability vector. Therefore, the reverse cross-entropy loss \( L_{RCE} \) maintains the performance on high-confidence samples. As a result, we find that the SCE keeps the gradient for high and low confidence predictions balanced, benefiting the optimization problem.

Thus, we now replace the CE loss from Eq. (1) by the symmetric cross-entropy, which is calculated using the softmax predictions of the teacher \( q_t^T \) and student \( p_t^T \) for the current test data \( x_t^T \). To further promote the consistency between the teacher and student for smaller perturbations, we additionally generate an augmented version of the current test data \( \tilde{x}_t^T = \text{Aug}(x_t^T) \) using the augmentations from [50]. \( \tilde{x}_t^T \) is then fed through the student network, which provides the softmax prediction \( \tilde{p}_t^T \). This prediction is subsequently used to calculate a second SCE loss \( L_{SCE}(q_t^T, \tilde{p}_t^T) \), resulting in the following self-training loss

\[
L_{ST}(x_t^T, \tilde{x}_t^T) = \frac{1}{4} \left( L_{SCE}(q_t^T, p_t^T) + L_{SCE}(q_t^T, \tilde{p}_t^T) \right). \quad (6)
\]

While it is common to only use the prediction of the teacher as the final output, we ensemble the predictions of both models by adding the student’s and teacher’s logits. This is motivated by the circumstance that the student can account for distribution shifts more quickly than the slower teacher. Although it is also possible to update the teacher faster, this would affect the teachers ability to stabilize the training.

**Mean teacher warm-up for more accurate predictions**

Since in the setting of online adaptation, it takes some time before weight-averaging results in more accurate predictions, we look into performing a mean teacher warm-up before deploying and adapting the model. Warm-up is conducted offline with the same batch size used during test-time by minimizing \( L_{SCE} \) for one epoch on 50,000 source
training samples. We want to note that no augmentation is applied during warm-up.

3.2. Contrastive Learning

The usage of contrastive learning in our approach is two-fold. On the one hand, it enables to further leverage the augmented test data to learn an invariance to small changes in the input space. On the other hand, the idea is to pull the test feature space towards the source domain where our source-pre-trained model is well-posed.

Before performing test-time adaptation, we first use the pre-trained student encoder at time step $t = 0$ to extract a prototype $r^S_c$ for each class $c$ in the source dataset. This is achieved by simply averaging all source features belonging to class $c$. Since the prototypes are kept fixed, no source data is required during test-time.

Now, given test feature $r^T_i = E_{enc}(x^T_i)$ extracted by the student encoder for the $i$-th test image contained in the current test batch with $N$ samples, we first compute the cosine similarity to each source prototype $r^S_c$. Then, the nearest source prototype is utilized to create a positive pair, which will later pull each test feature closer to the matching source class center. To further become invariant to small changes in the input space, each pair is extended with the corresponding features of the augmented test batch $x^T_i$. The batch now consists of $3N$ samples due to the test samples, the augmented view, and the nearest source prototypes. Let $i \in I := \{1, \ldots, 3N\}$, $A(i) := I \setminus \{i\}$, $V(i)$ be the different views for current sample $i$, and $z = \text{Proj}(r)$ denote the output of the non-linear projection of $r$; the contrastive loss is then defined as

$$L_{CL} = - \sum_{i \in I} \sum_{v \in V(i)} \log \frac{\exp(\text{sim}(z_i, z_o)/\tau)}{\sum_{a \in A(i)} \exp(\text{sim}(z_i, z_o)/\tau)},$$

where $\tau$ denotes the scalar temperature and $\text{sim}(u, v) = u^T v / (||u|| ||v||)$ is the cosine similarity.

3.3. Source Replay

Inspired by experience replay [22], rehearsal is a common technique in continual learning to keep a model in the same low-loss region from which learning was initiated while it is being updated on a new target distribution [47]. This is also desirable for continual test-time adaptation, since the model may again encounter samples originating from the source distribution or a closely related distribution during test-time. We further want to emphasize that self-training with noisy pseudo-labels can be prone to error accumulation. Even though the mean teacher in combination with the robust symmetric cross-entropy loss already addresses this issue, source replay can also be seen as a stabilizing component for the self-training process, potentially further preventing error accumulation.

To integrate source replay, we uniformly sample a labeled source batch $(x^S_i, y^S_i)$ and minimize

$$L^S_{CL}(x^S_i, y^S_i) = - \sum_{c=1}^C y^S_i \log p^S_{ic},$$

Clearly, using source replay during test-time requires to store at least parts of the source data in a buffer on the device. Since the buffer size can be a limiting factor, we investigate its influence on the performance later in the experiments. The overall loss function is given as

$$L(x^T_i, x^T_i, x^S_i, y^S_i) = L_{ST} + \lambda_{CL} L_{CL} + \lambda_{CE} L_{CE}^S,$$

where the gradient of the loss function is computed with respect to the student’s parameters $\theta_t$. The teacher $\theta_t'$ is updated by an exponential moving average.

4. Experiments

Datasets, metrics, and considered settings We evaluate our approach on CIFAR10C, CIFAR100C, and Imagenet-C, which were initially designed to benchmark robustness of classification networks [12]. All datasets include 15 different types of corruptions with 5 severity levels applied to the validation and test images of ImageNet and CIFAR, respectively [16]. To validate the effectiveness of our approach for domain shifts not caused by corruption, we additionally consider ImageNet-R [11], as well as DomainNet-126 [36], which is a subset of DomainNet [34]. While ImageNet-R contains 30,000 examples depicting different renditions of 200 ImageNet classes, DomainNet-126 has 126 classes and consists of four domains (real, clipart, painting, sketch).

We compare all methods in two different settings. First, we consider the continual benchmark, as introduced by [50]. Similar to the standard TTA setting used in [49], the continual benchmark also starts with an off-the-shelf model pre-trained on the source domain. However, while the standard TTA setting resets the model after it was adapted to a test domain, the continual setting does not assume to know when the domain changes. Therefore, the model is adapted to a sequence of test domains in an online fashion. In case of the corruption benchmark, the sequence consists of all 15 corruptions, each encountered at the highest severity level 5. For DomainNet-126, we randomly create four different domain sequences, with the only condition that every domain is used once as the source domain. More information about the DomainNet-126 benchmark and its benefits are located in Appendix C.

Since there are many applications where the domain shift does not occur abruptly but changes rather smoothly, we additionally consider the same gradual benchmark as in [26]. While the continual setting encounters each corruption at the highest severity level 5, the gradual setting increases the severity as follows: $1 \rightarrow 2 \rightarrow \cdots \rightarrow 5 \rightarrow \cdots \rightarrow 2 \rightarrow 1$. We report the error rate for all experiments.
Table 1. Classification error rate (%) for the CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNet-C online continual test-time adaptation task on the highest corruption severity level 5. For CIFAR10C the results are evaluated on WideResNet-28, for CIFAR100C on ResNeXt-29, and for Imagenet-C, ResNet-50 is used. We report the performance of our method averaged over 5 runs.

<table>
<thead>
<tr>
<th>Time</th>
<th>Method</th>
<th>Source-free Updates</th>
<th>Source BN–1</th>
<th>TENT-cont.</th>
<th>AdaContrast</th>
<th>CoTTA</th>
<th>GTTA-MIX</th>
<th>RMT (ours)</th>
<th>RMT (ours)</th>
<th>RMT (ours)</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Source only</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-3.7</td>
</tr>
<tr>
<td></td>
<td>BN–1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+15.0</td>
</tr>
<tr>
<td></td>
<td>TENT-cont.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+4.4</td>
</tr>
<tr>
<td></td>
<td>AdaContrast</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-2.7</td>
</tr>
<tr>
<td></td>
<td>CoTTA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-2.3</td>
</tr>
<tr>
<td></td>
<td>GTTA-MIX</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 2. Classification error rate (%) for the gradual CIFAR10-to-CIFAR10C, CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNet-C benchmark averaged over all 15 corruptions. We separately report the performance averaged over all severity levels (@ level 1–5) and averaged only over the highest severity level 5 (@ level 5). The number in brackets denotes the difference to the continual benchmark.

<table>
<thead>
<tr>
<th></th>
<th>Source-free Updates</th>
<th>Source BN–1</th>
<th>TENT-cont.</th>
<th>AdaContrast</th>
<th>CoTTA</th>
<th>GTTA-MIX</th>
<th>RMT</th>
<th>RMT</th>
<th>RMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10C @level 1–5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>@level 5</td>
<td>43.5</td>
<td>20.4</td>
<td>25.1 (+4.4)</td>
<td>15.8 (-2.7)</td>
<td>14.2 (-2.0)</td>
<td>13.0 (-2.6)</td>
<td>10.4 (-4.1)</td>
<td>9.4 (-4.5)</td>
</tr>
<tr>
<td>CIFAR100C @level 1–5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>23.6</td>
</tr>
<tr>
<td></td>
<td>@level 5</td>
<td>33.6</td>
<td>29.9</td>
<td>74.8 (+4.4)</td>
<td>33.0</td>
<td>26.3</td>
<td>24.7</td>
<td>26.4</td>
<td>24.2</td>
</tr>
<tr>
<td>Imagenet-C @level 1–5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>£</td>
<td>✓</td>
<td>✓</td>
<td>36.8</td>
</tr>
<tr>
<td></td>
<td>@level 5</td>
<td>58.4</td>
<td>48.3</td>
<td>64.6 (+3.7)</td>
<td>66.0</td>
<td>38.8</td>
<td>37.7</td>
<td>39.3</td>
<td>37.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>82.0</td>
<td>68.6</td>
<td>58.9 (+3.7)</td>
<td>72.6</td>
<td>43.1 (-9.4)</td>
<td>47.7 (-9.4)</td>
<td>41.5 (-18.3)</td>
<td>40.2 (-17.7)</td>
</tr>
</tbody>
</table>
Table 3. Classification error rate (%) for ImageNet-R and DomainNet-126 in the online continual TTA setting. We report the performance of our method averaged over 5 runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Source-free</th>
<th>Source-free</th>
<th>ImageNet-R</th>
<th>DomainNet-126</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>✓</td>
<td>✓</td>
<td>63.8</td>
<td>45.3</td>
</tr>
<tr>
<td>BN–1</td>
<td>✓</td>
<td>✓</td>
<td>60.4</td>
<td>45.1</td>
</tr>
<tr>
<td>TENT cont.</td>
<td>✓</td>
<td>✓</td>
<td>57.6</td>
<td>42.4</td>
</tr>
<tr>
<td>CoTTA</td>
<td>✓</td>
<td>✓</td>
<td>57.4</td>
<td>43.4</td>
</tr>
<tr>
<td>AdaContrast</td>
<td>✓</td>
<td>✓</td>
<td>59.1</td>
<td>37.8</td>
</tr>
<tr>
<td>GTTA-MIX</td>
<td>✓</td>
<td>✓</td>
<td>56.6</td>
<td>38.7</td>
</tr>
<tr>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>55.7</td>
<td>37.0</td>
</tr>
<tr>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>55.5</td>
<td>36.8</td>
</tr>
<tr>
<td>RMT (ours)</td>
<td>✓</td>
<td>✓</td>
<td>45.3</td>
<td>35.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>36.4</td>
<td>29.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
<td>32.8</td>
</tr>
</tbody>
</table>

Implementation details Following the RobustBench benchmark [6], a pre-trained WideResNet-28 [56] and ResNeXt-29 [54] is used for CIFAR10-to-CIFAR10C and CIFAR100-to-CIFAR100C, respectively. For ImageNet-to-Imagenet-C, ImageNet-R, and DomainNet-126, a source pre-trained ResNet-50 is applied. In the latter case, we use the same architecture and pre-trained weights as in [4]. We follow the implementation of [50], using the same hyperparameters. We weight all loss functions equally using $\lambda_{AC} = \lambda_{CE} = 1$ and set $\tau$ to the default value 0.1.

RMT variations Since each application has its own requirements in terms of efficiency, privacy, and memory, we introduce three variations of our method RMT. While the first variant omits source replay to account for situations where it is critical to store source data on the device, the latter two apply source replay but differ in the number of updates. Hence, they address the potential trade-off between efficiency and performance and are meant for applications where memory and computational power are not an issue.

Baselines To compare our method, we consider several source-free baselines, such as CoTTA [50], TENT continual [49], and AdaContrast [4]. In addition, we also compare to the non-source-free baseline GTTA-MIX [26] and the normalization-based method BN–1, which recalculates the batch normalization statistics using the current test batch.

4.1. Results for Continual Test-Time Adaptation

Domain shifts caused by corruption Table 1 shows the results for each corruption dataset in the continual setting. While the simple evaluation of the pre-trained source model (source only) leads to a high average error on all datasets, applying test-time normalization with BN–1 already drastically decreases the error rate without any error accumulation. This does not hold for TENT-continual, which suffers from heavy error accumulation for CIFAR100C, achieving an average error of 60.9%. Nevertheless, it performs on par and 6% better than BN–1 for CIFAR10C and Imagenet-C, respectively. Although it is always possible to use TENT-episodic, resetting the model after each update prevents the exploitation of previously learned knowledge, resulting in an equivalent performance to BN–1. CoTTA, on the other hand, is able to reduce the average error on most of the datasets without any signs of error accumulation. However, these results are achieved by applying heavy test-time augmentation, requiring up to 32 additional forward passes. If we now compare our source-free variant with CoTTA, the average error is significantly reduced. This variant even outperforms the non-source-free approach GTTA-MIX on CIFAR10C, while being only slightly worse on CIFAR100C. If access to source data is not an issue, the error rate can be further decreased. For applications, where the focus is less on efficiency and more on performance, the error rate can be further reduced by applying 4 update steps, as was also done by GTTA-MIX. Note that applying more update steps to source-free methods usually increases the error rate.

Natural domain shifts Table 3 shows the results for ImageNet-R and each sequence included in the continual DomainNet-126 benchmark. As depicted, all methods improve upon the non-adaptive source baseline. While CoTTA performs only slightly better than TENT continual in both settings, AdaContrast clearly takes the lead on DomainNet-126, while lacking performance on ImageNet-R. In contrast, our source-free approach sets new state-of-the-art results on both datasets and is even better than the non-source-
free approach GTTA-MIX. If we further leverage source replay, the error rate decreases again, reaching the best results when 4 update steps are applied.

4.2. Results for Gradual Test-Time Adaptation

Mean teachers are strong easy-to-hard learners In Tab. 2, we report the average error in the gradual setting across all severity levels and only with respect to level 5. This allows a direct comparison with the continual setting. While the performance of approaches like TENT continual and AdaContrast even degrades for some datasets, mean-teacher based approaches show a massive improvement of more than 18.3%. Hence, they can exploit the gradual shifts more effectively to reduce the error at level 5. Since a gradual increase in the severity level can also be seen as an easy-to-hard problem, we now revisit the continual setting, but sort the corruptions from low error to high error using the initial source model. Detailed results and the specific sequences are shown in Tab. 6 and 7 in the appendix. Again, we find that mean teachers are particularly well suited for easy-to-hard problems, where the error is 12% lower on ImageNet-C compared to an hard-to-easy sequence.

4.3. Single-Sample Test-Time Adaptation

Since timeliness can be important for some applications, we now consider single-sample TTA. A simple approach to overcome noisy gradients and poor estimates of the BN statistics caused by only having a single sample is to use a sliding window. In this case, the last b test samples are stored in a buffer. We only update the model weights every b steps, due to the correlation induced by the buffer. In the meantime, the entire buffer is forwarded to generate a prediction for the current test sample $x_{ti}^T$. Due to the smaller batch size, we decrease the learning rate by original batch size/b. Table 5 illustrates the results for single-sample TTA using various buffer sizes b. Due to the much smaller batch size used in this setting, the performance of the baseline BN–1 slightly degrades as the estimation of the batch statistics becomes more noisy. Although the performance of our approach is also slightly worse compared to the results obtained in the batch setting of TTA, the performance at a window size of 16 is still better or competitive to the state-of-the-art methods in the batch setting.

4.4. Ablation Studies

Component analysis First, we examine the effect of exploiting the symmetric cross-entropy loss $L_{SCE}$. As shown in Tab. 4, using a mean teacher with $L_{SCE}$ has a clear advantage over $L_{CE}$. If we further shortly warm up the mean teacher on the source domain using a linear learning rate increase, another significant reduction in error can be achieved on all datasets. Next, we carefully investigate our components. Utilizing our self-training loss $L_{ST}$ (A) in combination with the ensemble prediction (B) significantly improves the results compared to the mean teacher framework minimizing either the cross-entropy or the symmetric cross-entropy. While extending our approach with a contrastive component (C) further reduces the average error for all datasets, adding warm-up (D) and source replay (E) again substantially improves the overall performance.

Ablations Additional investigations concerning the effect of different numbers of update steps, various amounts of saved source samples, and a sensitivity analysis with respect to the temperature $\tau$ and the momentum term $\alpha$ are shown in Tab. 9 in the appendix. While we find that 2 and 4 update steps provide a good balance between performance and computational complexity, RMT profits from even more update steps. Although even our source-free variant already sets new standards on all benchmarks, having access to only 1% of the source data during test-time is beneficial.

5. Conclusion

In this work, we showed that a mean teacher with a symmetric cross-entropy loss combined with contrastive learning sets a new standard in the area of continual and gradual TTA. We motivate the usage of a symmetric cross-entropy loss by analyzing the respective gradient properties. We achieve state-of-the-art results on all common benchmarks and introduced a new benchmark based on DomainNet-126 to further demonstrate the effectiveness for a larger variety of domain shifts. In case privacy or accessibility is no concern, replaying a small percentage of source data improves the performance and allows to perform multiple update steps, resulting in an additional performance gain.

Acknowledgments This publication was created as part of the research project "KI Delta Learning" (project number: 19A19013R) funded by BMWi.
References


[42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems, 30, 2017. 2, 3


[53] Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. Ace: adapting to changing environments


