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.
[1] Code 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 [39] during training improves generalizability and robustness [11, 13, 19]. Another direction is to learn domain-invariant features, as addressed by [7, 32]. The idea of domain randomization [43, 45] 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 [52] 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 [9, 46], discrepancy based loss functions [3, 40, 55], or contrastive learning [15, 25]. While it is also possible to adapt the domains in the input space [14, 27, 37, 53], 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 [23, 29, 48, 58]. To increase the reliability of the network's predictions, [8, 44] rely on a mean teacher [42].

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, [38] 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 [20] 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, [49] minimizes the entropy with respect to the BN parameters. [57] minimizes the entropy with respect to all parameters and uses test-time augmentation [16] to artificially increase the batch size. Other methods apply contrastive learning [4] or even introduce an additional self-supervision loss during source pre-training that is later leveraged to perform the adaptation during test-time [1, 2, 24, 41]. Diversity regularizers [21, 33] 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 [2, 10, 31, 57].

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 [49], but methods solely based on self-training can be prone to error accumulation. This is a result of miscalibrated predictions, as highlighted in [50]. CoTTA [50] 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 [28]. [26] proposes a method that not only considers a continual setting, but also investigates scenarios where shifts are gradual. Based on the theory of [17] that self-training is sufficient as long as the encountered shifts are small enough, GTTA [26] introduces intermediate domains by mixup or style transfer for effective self-training.

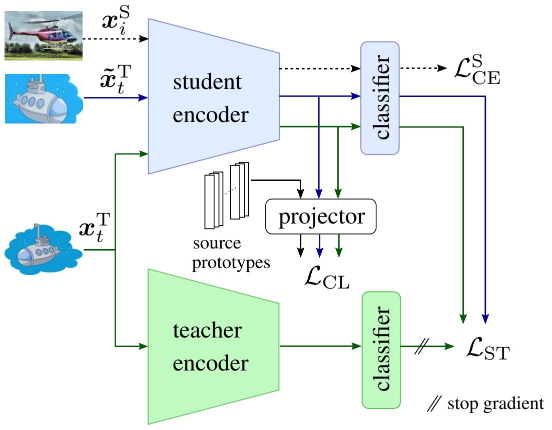

Figure 1. **RMT framework:** Before the adaptation, the student and teacher networks are initialized with source pre-trained weights. Source prototypes are extracted for each class and a mean teacher warm-up is performed. During test-time, the current test batch $\boldsymbol{x}_t^{\mathrm{T}}$ and an augmented version $\tilde{\boldsymbol{x}}_t^{\mathrm{T}}$ are encoded by the student. Test features, augmented test features, and the nearest source prototypes based on the cosine similarity are used for the contrastive loss $\mathcal{L}_{\mathrm{CL}}$. Self-training is performed by using two symmetric cross-entropy losses. If source data is available, we uniformly sample a batch $(\boldsymbol{x}_i^{\mathrm{S}}, \boldsymbol{y}_i^{\mathrm{S}})$ from the source data and compute the cross-entropy loss $\mathcal{L}_{\mathrm{CE}}^{\mathrm{S}}$. The student is updated by minimizing the previously mentioned losses. The teacher is updated via an exponential moving average of the student's parameters.

3. Methodology

In many practical applications, the environmental conditions change over time. Hence, a model $f_{\boldsymbol{\theta}_0}$, pre-trained on source data $(\mathcal{X}^{\mathrm{S}}, \mathcal{Y}^{\mathrm{S}})$, can quickly become sub-optimal and provide only inadequate predictions for the current test data $\boldsymbol{x}_t^{\mathrm{T}}$ 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 $\boldsymbol{x}_t^{\mathrm{T}}$. Depending on the application, labeled source data $(\mathcal{X}^{\mathrm{S}}, \mathcal{Y}^{\mathrm{S}})$ 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 $\boldsymbol{\theta}_0$. During test-time, the student $f_{\boldsymbol{\theta}_t}$ is updated ($\boldsymbol{\theta}_t \to \boldsymbol{\theta}_{t+1}$) by minimizing the cross-entropy (CE)

$$\mathcal{L}_{\mathrm{CE}}(\boldsymbol{q}, \boldsymbol{p}) = -\sum_{c=1}^{C} q_c \log p_c \tag{1}$$

between the softmax predictions of the student $\boldsymbol{p}$ and the teacher $\boldsymbol{q}$, with C being the total number of classes. Subsequently, the non-trainable weights of the teacher $\boldsymbol{\theta}_t'$ are updated using an exponential moving average $\boldsymbol{\theta}_{t+1}' = \alpha\boldsymbol{\theta}_t' + (1-\alpha)\boldsymbol{\theta}_{t+1}$, where α 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

$$\frac{\partial \mathcal{L}_{\mathrm{CE}}}{\partial p} = \frac{p-q}{p-p^2}, \tag{2}$$

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

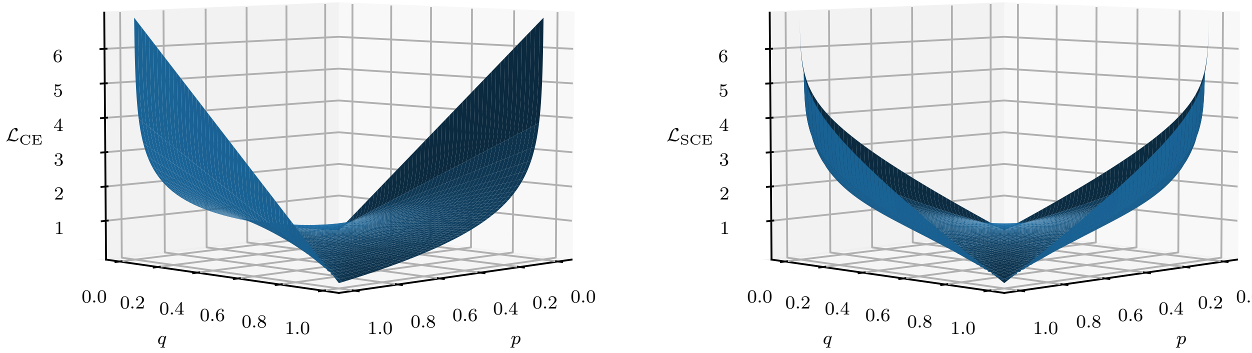

Figure 2. Loss surface illustrated for the binary case of the cross-entropy loss $\mathcal{L}_{\text{CE}}$ and the symmetric cross-entropy $\mathcal{L}_{\text{SCE}}$ in dependence of the confidences p and q of the student and teacher, respectively.

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 $\boldsymbol{q}$ and $\boldsymbol{p}$, the symmetric cross-entropy is defined as

$$\mathcal{L}_{\text{SCE}}(\boldsymbol{q}, \boldsymbol{p}) = -\sum_{c=1}^{C} q_c \log p_c - \sum_{c=1}^{C} p_c \log q_c, \tag{3}$$

where the first term corresponds to the cross-entropy loss $\mathcal{L}_{\text{CE}}$, while the second term is the reverse cross-entropy (RCE) loss $\mathcal{L}_{\text{RCE}}$.

For the binary SCE loss, the gradient is given by

$$\frac{\partial \mathcal{L}_{\text{SCE}}}{\partial p} = \frac{\partial \mathcal{L}_{\text{CE}}}{\partial p} + \log(1/q - 1). \tag{4}$$

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 $\mathcal{L}_{\text{RCE}}$ with respect to the j-th output element of the network is given by

$$\frac{\partial \mathcal{L}_{\text{RCE}}}{\partial z_j} = p_j \Big(\sum_{c=1}^{C} p_c \log q_c - \log q_j \Big). \tag{5}$$

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 $\mathcal{L}_{\text{RCE}}$ 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 $\boldsymbol{q}_t^{\text{T}}$ and student $\boldsymbol{p}_t^{\text{T}}$ for the current test data $\boldsymbol{x}_t^{\text{T}}$. To further promote the consistency between the teacher and student for smaller perturbations, we additionally generate an augmented version of the current test data $\tilde{\boldsymbol{x}}_t^{\text{T}} = \text{Aug}(\boldsymbol{x}_t^{\text{T}})$ using the augmentations from [50]. $\tilde{\boldsymbol{x}}_t^{\text{T}}$ is then fed through the student network, which provides the softmax prediction $\tilde{\boldsymbol{p}}_t^{\text{T}}$. This prediction is subsequently used to calculate a second SCE loss $\mathcal{L}_{\text{SCE}}(\boldsymbol{q}_t^{\text{T}}, \tilde{\boldsymbol{p}}_t^{\text{T}})$, resulting in the following self-training loss

$$\mathcal{L}_{\text{ST}}(\boldsymbol{x}_t^{\text{T}}, \tilde{\boldsymbol{x}}_t^{\text{T}}) = \frac{1}{4} \Big(\mathcal{L}_{\text{SCE}}(\boldsymbol{q}_t^{\text{T}}, \boldsymbol{p}_t^{\text{T}}) + \mathcal{L}_{\text{SCE}}(\boldsymbol{q}_t^{\text{T}}, \tilde{\boldsymbol{p}}_t^{\text{T}}) \Big). \tag{6}$$

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 $\mathcal{L}_{\text{SCE}}$ 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 twofold. 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 $\boldsymbol{r}_c^{\mathrm{S}}$ 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 $\boldsymbol{r}_{ti}^{\mathrm{T}} = \mathrm{Enc}(\boldsymbol{x}_{ti}^{\mathrm{T}})$ 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 $\boldsymbol{r}_c^{\mathrm{S}}$. 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 $\tilde{\boldsymbol{x}}_t^{\mathrm{T}}$. The batch now consists of $3N$ samples due to the test samples, the augmented view, and the nearest source prototypes. Let $i \in I := \{1, \ldots, 3N\}$, $A(i) := I \backslash \{i\}$, $V(i)$ be the different views for current sample i, and $\boldsymbol{z} = Proj(\boldsymbol{r})$ denote the output of the non-linear projection of $\boldsymbol{r}$, the contrastive loss is then defined as

$$\mathcal{L}_{\mathrm{CL}} = -\sum_{i \in I} \sum_{v \in V(i)} \log \frac{\exp\big(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_v)/\tau\big)}{\sum\limits_{a \in A(i)} \exp\big(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_a)/\tau\big)}, \quad (7)$$

where τ denotes the scalar temperature and $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T \boldsymbol{v} / (\|\boldsymbol{u}\| \|\boldsymbol{v}\|)$ 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 $(\boldsymbol{x}_i^{\mathrm{S}}, \boldsymbol{y}_i^{\mathrm{S}})$ and minimize

$$\mathcal{L}_{\mathrm{CE}}^{\mathrm{S}}(\boldsymbol{x}_i^{\mathrm{S}}, \boldsymbol{y}_i^{\mathrm{S}}) = -\sum_{c=1}^{C} y_{ic}^{\mathrm{S}} \log p_{ic}^{\mathrm{S}}. \quad (8)$$

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

$$\mathcal{L}(\boldsymbol{x}_t^{\mathrm{T}}, \tilde{\boldsymbol{x}}_t^{\mathrm{T}}, \boldsymbol{x}_i^{\mathrm{S}}, \boldsymbol{y}_i^{\mathrm{S}}) = \mathcal{L}_{\mathrm{ST}} + \lambda_{\mathrm{CL}} \mathcal{L}_{\mathrm{CL}} + \lambda_{\mathrm{CE}} \mathcal{L}_{\mathrm{CE}}^{\mathrm{S}}, \quad (9)$$

where the gradient of the loss function is computed with respect to the student's parameters $\boldsymbol{\theta}_t$. The teacher $\boldsymbol{\theta}_t'$ 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: $1 \to 2 \to \cdots \to 5 \to \cdots \to 2 \to 1$. 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.

	Time			$t \longrightarrow$															
	Method	Source-free	Updates	*Gaussian*	*shot*	*impulse*	*defocus*	*glass*	*motion*	*zoom*	*snow*	*frost*	*fog*	*brightness*	*contrast*	*elastic*	*pixelate*	*jpeg*	Mean
CIFAR10C	Source only	✓	-	72.3	65.7	72.9	46.9	54.3	34.8	42.0	25.1	41.3	26.0	9.3	46.7	26.6	58.5	30.3	43.5
	BN–1	✓	-	28.1	26.1	36.3	12.8	35.3	14.2	12.1	17.3	17.4	15.3	8.4	12.6	23.8	19.7	27.3	20.4
	TENT-cont.	✓	1	24.8	20.6	28.6	14.4	31.1	16.5	14.1	19.1	18.6	18.6	12.2	20.3	25.7	20.8	24.9	20.7
	AdaContrast	✓	1	29.1	22.5	30.0	14.0	32.7	14.1	12.0	16.6	14.9	14.4	8.1	10.0	21.9	17.7	20.0	18.5
	CoTTA	✓	1	24.3	21.3	26.6	11.6	27.6	12.2	10.3	14.8	14.1	12.4	7.5	10.6	18.3	13.4	17.3	16.2
	GTTA-MIX	✗	4	23.4	18.3	25.5	**10.1**	27.3	11.6	10.1	14.1	13.0	10.9	7.4	9.0	19.4	14.5	19.8	15.6
	RMT (ours)	✓	1	21.9	18.6	24.1	10.8	23.6	12.0	10.4	13.0	12.4	11.4	8.3	10.1	15.2	11.3	14.6	14.5±0.09
	RMT (ours)	✗	1	21.7	18.6	24.2	10.3	24.0	11.2	9.5	12.1	11.7	10.3	**7.0**	8.7	14.8	10.5	14.5	13.9±0.07
	RMT (ours)	✗	4	**20.8**	**16.5**	**20.5**	10.4	**20.1**	**10.8**	**9.2**	**11.0**	**10.4**	**9.7**	7.3	**8.0**	**12.3**	**8.7**	**11.6**	**12.5**±0.07
CIFAR100C	Source only	✓	-	73.0	68.0	39.4	29.3	54.1	30.8	28.8	39.5	45.8	50.3	29.5	55.1	37.2	74.7	41.2	46.4
	BN–1	✓	-	42.1	40.7	42.7	27.6	41.9	29.7	27.9	34.9	35.0	41.5	26.5	30.3	35.7	32.9	41.2	35.4
	TENT-cont.	✓	1	37.2	35.8	41.7	37.9	51.2	48.3	48.5	58.4	63.7	71.1	70.4	82.3	88.0	88.5	90.4	60.9
	AdaContrast	✓	1	42.3	36.8	38.6	27.7	40.1	29.1	27.5	32.9	30.7	38.2	25.9	28.3	33.9	33.3	36.2	33.4
	CoTTA	✓	1	40.1	37.7	39.7	26.9	38.0	27.9	26.4	32.8	31.8	40.3	24.7	26.9	32.5	28.3	33.5	32.5
	GTTA-MIX	✗	4	36.4	**32.1**	34.0	**24.4**	35.2	25.9	23.9	28.9	27.5	30.9	22.6	**23.4**	29.4	25.5	33.3	28.9
	RMT (ours)	✓	1	38.5	34.4	35.4	26.4	32.7	27.0	25.0	27.5	27.6	30.0	24.0	25.8	27.0	25.2	28.4	29.0±0.17
	RMT (ours)	✗	1	37.4	33.8	34.3	24.8	32.0	25.3	**23.6**	26.2	26.2	28.9	**21.9**	23.5	25.4	**23.2**	27.4	27.6±0.04
	RMT (ours)	✗	4	**36.2**	32.2	**32.1**	25.0	**29.8**	**25.0**	**23.6**	**25.4**	**25.2**	**27.1**	23.1	**23.4**	**24.4**	23.4	**25.9**	**26.8**±0.08
ImageNet-C	Source only	✓	-	97.8	97.1	98.2	81.7	89.8	85.2	78.0	83.5	77.1	75.9	41.3	94.5	82.5	79.3	68.6	82.0
	BN–1	✓	-	85.0	83.7	85.0	84.7	84.3	73.7	61.2	66.0	68.2	52.1	34.9	82.7	55.9	51.3	59.8	68.6
	TENT-cont.	✓	1	81.6	74.6	72.7	77.6	73.8	65.5	55.3	61.6	63.0	51.7	38.2	72.1	50.8	47.4	53.3	62.6
	AdaContrast	✓	1	82.9	80.9	78.4	81.4	78.7	72.9	64.0	63.5	64.5	53.5	38.4	66.7	54.6	49.4	53.0	65.5
	CoTTA	✓	1	84.7	82.1	80.6	81.3	79.0	68.6	57.5	60.3	60.5	48.3	36.6	66.1	47.2	41.2	46.0	62.7
	GTTA-MIX	✗	4	75.2	**67.4**	**64.6**	73.3	72.5	61.8	**52.7**	**53.0**	**54.9**	**42.6**	**33.8**	63.9	48.9	44.4	47.0	57.1
	RMT (ours)	✓	1	77.9	73.1	69.9	73.5	71.1	63.1	57.1	57.1	59.2	50.4	42.9	60.1	49.0	45.7	46.9	59.8±0.18
	RMT (ours)	✗	1	77.3	73.2	71.1	73.1	71.2	61.2	53.7	54.3	58.0	46.1	38.2	58.5	**45.4**	**42.3**	**44.5**	57.9±0.26
	RMT (ours)	✗	4	**74.8**	68.6	65.2	**68.2**	**66.2**	**59.0**	53.4	53.7	56.9	47.5	41.2	**54.1**	46.0	44.6	45.9	**56.4**±0.25

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.

		Source	BN–1	TENT-cont.	AdaCont.	CoTTA	GTTA-MIX	RMT	RMT	RMT
	Source-free	✓	✓	✓	✓	✓	✗	✓	✗	✗
	Updates	-	-	1	1	1	4	1	1	4
CIFAR10C	@level 1–5	24.7	13.7	20.4	12.1	10.9	11.8	9.3	**8.1**	8.6
	@level 5	43.5	20.4	25.1 (+4.4)	15.8 (-2.7)	14.2 (-2.0)	13.0 (-2.6)	10.4 (-4.1)	9.4 (-4.5)	**9.0** (-3.5)
CIFAR100C	@level 1–5	33.6	29.9	74.8	33.0	26.3	24.7	26.4	**23.6**	24.2
	@level 5	46.4	35.4	75.9 (+15.0)	35.9 (+2.5)	28.3 (-4.2)	26.1 (-2.8)	26.9 (-2.1)	**24.3** (-3.2)	24.5 (-2.3)
Imagenet-C	@level 1–5	58.4	48.3	46.4	66.3	38.8	37.7	39.3	37.8	**36.8**
	@level 5	82.0	68.6	58.9 (-3.7)	72.6 (+7.1)	43.1 (-19.6)	47.7 (-9.4)	41.5 (-18.3)	40.2 (-17.7)	**37.5** (-18.9)

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.

				DomainNet-126				
Method	Source-free	Updates	*ImageNet-R*	*real* →	*clipart* →	*painting* →	*sketch* →	Mean
Source only	✓	-	63.8	45.3	49.3	41.7	44.8	45.3
BN–1	✓	-	60.4	45.1	45.2	39.5	37.8	41.9
TENT cont.	✓	1	57.6	42.4	44.2	37.2	37.5	40.3
CoTTA	✓	1	57.4	43.4	43.0	36.4	36.3	39.8
AdaContrast	✓	1	59.1	37.8	37.6	32.3	31.9	34.9
GTTA-MIX	✗	4	56.6	38.7	42.4	33.6	34.2	37.2
RMT (ours)	✓	1	55.7	37.0	37.9	31.7	32.1	34.7
RMT (ours)	✗	1	55.5	36.8	37.1	30.6	31.1	33.9
RMT (ours)	✗	4	**53.5**	**35.1**	**36.4**	**29.9**	**29.9**	**32.8**

Table 4. Classification error rate (%) for different configurations averaged over 3 runs.

Method	*CIFAR10C*	*CIFAR100C*	*ImageNet-C*	*ImageNet-R*	*DomNet-126*	Mean
MT ($\mathcal{L}_{\text{CE}}$)	18.8	32.1	65.9	59.6	41.9	43.7
MT ($\mathcal{L}_{\text{SCE}}$)	17.9	31.5	62.8	57.3	39.7	41.8
+ warm-up	16.7	30.6	61.3	55.0	38.8	40.5
A $\mathcal{L}_{\text{ST}}$	18.0	31.2	61.9	57.2	39.1	41.5
B + ensemble	17.1	30.5	60.1	56.5	36.8	40.2
C + contrastive	16.7	30.1	59.9	55.6	35.0	39.5
D + warm-up	14.5	29.0	59.8	55.7	34.7	38.7
E + src. replay	**13.9**	**27.6**	**57.9**	**55.5**	**33.9**	**37.8**

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 $\lambda_{\text{CL}} = \lambda_{\text{CE}} = 1$ and set τ 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-

Table 5. Classification error rate (%) for single-sample TTA.

Method	Window size	*CIFAR10C*	*CIFAR100C*	*ImageNet-C*	*ImageNet-R*	*DomNet-126*	Mean
Source only	-	43.5	46.4	82.0	63.8	45.3	56.2
BN–1	8	26.3	43.8	74.6	64.7	49.7	51.8
BN–1	16	23.2	39.5	71.0	62.2	45.4	48.3
BN–1	32	21.9	37.4	69.3	60.7	43.4	46.5
RMT (ours)	8	16.7	33.6	72.0	59.7	43.7	45.1
RMT (ours)	16	15.2	30.8	63.9	57.8	36.8	40.9
RMT (ours)	32	14.3	28.1	59.9	56.1	35.3	38.7

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 $\boldsymbol{x}_{ti}^{\mathrm{T}}$. Due to the smaller batch size, we decrease the learning rate by $\text{original batch size}/b$. 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 $\mathcal{L}_{\text{SCE}}$. As shown in Tab. 4, using a mean teacher with $\mathcal{L}_{\text{SCE}}$ has a clear advantage over $\mathcal{L}_{\text{CE}}$. 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 $\mathcal{L}_{\text{ST}}$ (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 τ and the momentum term α 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

[1] Alexander Bartler, Florian Bender, Felix Wiewel, and Bin Yang. Ttaps: Test-time adaption by aligning prototypes using self-supervision. *arXiv preprint arXiv:2205.08731*, 2022. 2

[2] Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pages 3080–3090. PMLR, 2022. 2

[3] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 3422–3429, 2020. 2

[4] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022. 2, 7

[5] Tong Chu, Yahao Liu, Jinhong Deng, Wen Li, and Lixin Duan. Denoised maximum classifier discrepancy for source-free unsupervised domain adaptation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*, volume 2, 2022. 4

[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

[7] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[8] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018. 2

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016. 2

[10] Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven test-time adaptation. *arXiv preprint arXiv:2207.03442*, 2022. 2

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 1, 2, 5

[12] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 5

[13] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 1, 2

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 2

[15] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 2

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5

[17] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*, pages 5468–5479. PMLR, 2020. 1, 2

[18] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020. 3

[19] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12383–12392, 2021. 2

[20] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018. 2

[21] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020. 2

[22] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 8(3):293–321, 1992. 5

[23] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021. 2

[24] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2

[25] Robert A Marsden, Alexander Bartler, Mario Döbler, and Bin Yang. Contrastive learning and self-training for unsupervised domain adaptation in semantic segmentation. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 2

[26] Robert A Marsden, Mario Döbler, and Bin Yang. Gradual test-time adaptation by self-training and style transfer. *arXiv preprint arXiv:2208.07736*, 2022. 1, 2, 5, 7

[27] Robert A Marsden, Felix Wiewel, Mario Döbler, Yang Yang, and Bin Yang. Continual unsupervised domain adaptation for semantic segmentation using a class-specific transfer. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 2

[28] 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. 2

[29] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. *arXiv preprint arXiv:2008.12197*, 2020. 2

[30] Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[31] M Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14765–14775, 2022. 2

[32] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18. PMLR, 2013. 1, 2

[33] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik Metzen. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021. 2, 4

[34] 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/CVF international conference on computer vision*, pages 1406–1415, 2019. 5

[35] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 1

[36] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8050–8058, 2019. 5

[37] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8503–8512, 2018. 2

[38] 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. 1, 2

[39] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. 2

[40] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016. 2

[41] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020. 2

[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

[43] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017. 1, 2

[44] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1379–1389, 2021. 2

[45] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018. 1, 2

[46] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 2

[47] Eli Verwimp, Matthias De Lange, and Tinne Tuytelaars. Rehearsal revealed: The limits and merits of revisiting samples in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9385–9394, 2021. 5

[48] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2517–2526, 2019. 2

[49] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 1, 2, 5, 7

[50] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 1, 2, 4, 5, 7

[51] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019. 3, 4

[52] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 2

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

for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2121–2130, 2019. 2

[54] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 7

[55] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2272–2281, 2017. 2

[56] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 7

[57] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021. 2

[58] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. 2