

# Universal Test-time Adaptation through Weight Ensembling, Diversity Weighting, and Prior Correction

Robert A. Marsden \*    Mario Döbler \*    Bin Yang

University of Stuttgart

frobert.marsden, mario.doebler, bin.yang@iss.uni-stuttgart.de

## Abstract

*Since distribution shifts are likely to occur during test-time and can drastically decrease the model's performance, online test-time adaptation (TTA) continues to update the model after deployment, leveraging the current test data. Clearly, a method proposed for online TTA has to perform well for all kinds of environmental conditions. By introducing the variable factors domain non-stationarity and temporal correlation, we first unfold all practically relevant settings and define the entity as universal TTA. We want to highlight that this is the first work that covers such a broad spectrum, which is indispensable for the use in practice. To tackle the problem of universal TTA, we identify and highlight several challenges a self-training based method has to deal with: 1) model bias and the occurrence of trivial solutions when performing entropy minimization on varying sequence lengths with and without multiple domain shifts, 2) loss of generalization which exacerbates the adaptation to multiple domain shifts and the occurrence of catastrophic forgetting, and 3) performance degradation due to shifts in class prior. To prevent the model from becoming biased, we leverage a dataset and model-agnostic certainty and diversity weighting. In order to maintain generalization and prevent catastrophic forgetting, we propose to continually weight-average the source and adapted model. To compensate for disparities in the class prior during test-time, we propose an adaptive prior correction scheme that reweights the model's predictions. We evaluate our approach, named ROID, on a wide range of settings, datasets, and models, setting new standards in the field of universal TTA. Code is available at: <https://github.com/mario-doebler/test-time-adaptation>*

## 1. Introduction

Deep neural networks achieve remarkable performance, as long as training and test data originate from the same dis-

tribution. However, in the real world, environmental changes can occur during test-time and will likely degrade the performance of the deployed model. Domain generalization aims to address potential domain shifts by improving the robustness and generalization of the model directly during training [12, 14, 31, 46, 48]. Due to the wide range of data shifts [36] which are typically unknown during training [29], the effectiveness of these approaches remains limited. Since the test data provide insights into the current distribution shift, online test-time adaptation (TTA) emerged. In TTA, the model is adapted directly during test-time using an unsupervised loss function like the entropy and the available test sample(s) at time step  $t$ .

Although TENT [51] has demonstrated success in adapting to single domain shifts, recent research on TTA has identified more challenging scenarios where methods solely based on self-training, such as TENT, often fail [2, 11, 34, 53, 60]. However, these studies again have predominantly focused on specific settings, overlooking the broad spectrum of possible scenarios. Therefore, we initiate our approach by identifying two key factors that encompass all practically relevant scenarios: *domain non-stationarity* and *temporal correlation*. We denote the complete set of scenarios, including the capability to adapt to arbitrary domains, as *universal TTA*, illustrated in Figure 1 a).

In the following, we highlight the challenges imposed by these environmental factors and derive design choices for our framework ROID. Starting with the simplest scenario of adapting to a single domain with i.i.d. data, we empirically show that even when encountering a uniform class distribution a self-training based approach is likely to develop a bias towards certain classes. This poses the risk that when adapting to long sequences, a model collapse is likely, where finally only a small subset or a single class is predicted. Therefore, maintaining diverse predictions is essential. To address this, we introduce a dataset and model-agnostic certainty and diversity loss weighting.

Considering the degree of *domain non-stationarity*, common scenarios range from gradual or continual domain shifts [25, 53] to consecutive test samples originating from

---

\*Equal contribution.

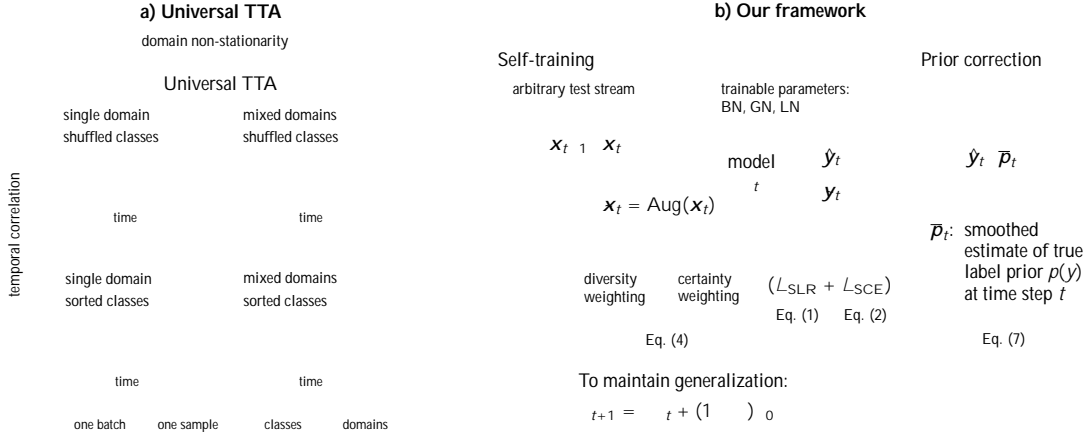


Figure 1. Illustration of universal TTA for a single or a batch of test samples and our framework ROID.

different domains. To deal with non-stationarity, maintaining diversity is even more crucial. We empirically show that the presence of multiple domain shifts can explicitly trigger a collapse to a trivial solution. In contrast to the single domain scenario, continual TTA [53] considers the adaptation to a sequence of multiple domains. In this context, in order to ensure effective adaptation to future shifts, a model must uphold its generalization. We hypothesize that adapting a model through self-training on a narrow distribution deteriorates generalization. This is validated by our empirical observations, indicating that a stronger adaptation results in a higher generalization error and promotes catastrophic forgetting. In response, we propose to continually weight-average the current model with the initial source model and denote this as weight ensembling. Dealing with mixed domains presents additional difficulties, such as adapting to multiple target domains simultaneously and the ineffectiveness of covariate shift mitigation through recalculating the batch normalization (BN) statistics during test-time [42].

In case of *temporally correlated* data or single sample TTA, the estimation of reliable BN statistics is not possible. While introducing a buffer can mitigate this problem [60], it can raise privacy and memory issues. Alternatively, one can leverage normalization layers like group normalization (GN) or layer normalization (LN), which do not require a batch of data to estimate the statistic and are thus better suited [34, 44]. Since applying diversity weighting promotes the model output to be unbiased, i.e., approximately uniform, even a model that is well adapted to the current domain shift will underperform in a temporally correlated setting. This is due to the existing shift in the class prior. Therefore, instead of allowing the model to become biased, we propose prior correction which introduces an adaptive additive smoothing scheme to reweight the model’s predictions.

We summarize our contributions as follows: 1) Our proposed method significantly outperforms existing approaches in the challenging setting of universal TTA. This indicates the potential of our method to be used in practical scenarios.

2) Through our analysis, we provide valuable insights into the challenges that arise when models are subjected to self-training during test-time. 3) Depending on the application, single-sample TTA might be of interest. We highlight that architectures that do not rely on batch normalization layers allow to recover the batch TTA setting from a single sample scenario by doing gradient accumulation. This also dramatically reduces memory consumption. 4) We show that current methods, even if proposed for challenging settings, often fail to fully address the whole picture of universal TTA—a result of our extensive and broad experiments in terms of settings, domain shifts, and models.

## 2. Related Work

**Unsupervised domain adaptation** Since domain generalization has its limitations due to the high amount of possible domain shifts that are unknown during training, in the field of unsupervised domain adaptation (UDA) [55], labeled source and unlabeled target data are used to adapt to the target domain. One line of work minimizes the discrepancy between the source and target feature distribution by either using adversarial learning [9, 49], discrepancy based loss functions [3, 43, 59], or contrastive learning [17, 24]. Instead of aligning the feature space, it is also possible to align the input space [15, 26, 41, 57], e.g., via style-transfer. Recently, self-training based approaches have shown to be powerful. Self-training uses the networks’ predictions on the target domain as pseudo-labels to minimize, e.g., a (cross-)entropy loss [21, 28, 50, 64]. Often filtering pseudo-labels is applied to remove unreliable samples. Mean teachers [45] can be further leveraged to increase the reliability of the network’s predictions [8, 47].

**Test-time adaptation** While UDA typically performs offline model adaptation, online test-time adaptation adapts the model to an unknown domain shift directly during inference using the currently available test samples. [42] showed that estimating new batch normalization (BN) statistics during test-time can significantly improve the performance on shifts

caused by corruptions. While only updating the BN statistics is computationally efficient, it has its limitations, especially when it comes to natural domain shifts. Therefore, recent TTA methods further update the model weights by relying on self-training. TENT [51] demonstrated that minimizing the entropy with respect to the batch normalization parameters can be successful for single-target adaptation. EATA [33] extends this idea by weighting the samples according to their reliability and diversity. Further, they use elastic weight consolidation [18] to prevent catastrophic forgetting [27] on the initial training domain. However, this requires access to data from the initial training domain, which is not always available in practice. To circumvent a model collapse to trivial solutions caused by confidence maximization, [20, 32] make use of diversity regularizers. Contrastive learning has also found its application in TTA [4, 5].

While some TTA methods only consider the adaptation to a single domain, in the real world, it is common to encounter multiple domain shifts. Therefore, [53] introduced the setting of continual test-time adaptation, where a model has to adapt to a sequence of different domains. While self-training based methods such as [51] can be applied to the continual setting, they can be prone to error accumulation [53]. To prevent error accumulation, [53] proposes to use weight and augmentation-averaged predictions in combination with a stochastic restore to mitigate catastrophic forgetting. RMT [5] proposes a robust mean teacher to deal with multiple domain shifts and GTTA [25] uses mixup and style-transfer to artificially create intermediate domains. LAME [2], NOTE [11], SAR [34], and RoTTA [60] propose methods that focus on dealing with temporally correlated data. While LAME only adapts the model’s output with Laplacian adjusted maximum-likelihood estimation, NOTE and RoTTA introduce a buffer to simulate an i.i.d. stream. SAR proposes a sharpness-aware and reliable entropy minimization method to be robust to large and noisy gradients.

Further areas of test-time adaptation focus on settings where the collection of a batch of data may not be feasible due to timeliness. Methods for single-sample TTA [1, 10, 30, 62] often rely on artificially creating a batch of data through test-time augmentation [19], which drastically increases the computational overhead. Due to only using a single sample for adapting the model, updates can be noisy and therefore the adaptation capability may be limited. Further, the area of test-time training modifies the initial pre-training phase by introducing an additional self-supervision loss that is also exploited to adapt the model during test-time [1, 22, 44]. Thus, test-time training is unable to use any off-the-shelf pre-trained model.

### 3. Self-training for Test-time Adaptation

Let  $\theta_0$  denote the weights of a deep neural network pre-trained on labeled source data  $(X; Y)$ . While the network

will typically perform well on data originating from the same domain, this is usually not the case when the model encounters data from different domains. This lack of generalization to out of distribution data is a problem in practice since the environmental conditions are likely to change from time to time. To keep the networks’ performance high during inference, online test-time adaptation continues to update the model after deployment using an unsupervised loss function like the entropy and the currently available test data  $\mathbf{x}_t$  at time step  $t$ .

**Test-time adaption through self-training carries the risk of generalization loss** Adapting a model to a target domain effectively means moving the model from its initial source parameterization to a parameterization that better models the current target distribution. This carries the risk that predictions on the source distribution become inaccurate, but also carries the risk of losing generalization when the target distribution is narrow. The former is known as catastrophic forgetting. We now want to highlight the latter, since generalization is a so far underestimated topic in TTA and is important for coping with non-stationary domains.

To study the impact of performing entropy minimization on generalization, we consider a typical TTA framework (TENT) where only parameters of the BN layers are trained while the rest remains frozen. We utilize an ImageNet pre-trained ResNet-50 and adapt the model using 40,000 samples of one of the corruptions from ImageNet-C [13]. To investigate the adaptation and generalization, we then evaluate the adapted model for each corruption on the remaining 10,000 samples. In Figure 2, we illustrate the difference of error for a moderate and a stronger adaptation, corresponding to a learning rate of  $10^{-4}$  and  $10^{-3}$ , respectively. As one would expect, a stronger adaptation leads to an improvement for samples originating from the same or a similar domain. However, this comes with the drawback that the performance on other domains deteriorates, indicating a loss of generalization. As a result, adapting to future domains is hindered. The same effect can be observed for the source domain, depicted in the last column, showing signs of catastrophic forgetting. As illustrated in Figure 5 located in Appendix A.1, the effect also occurs for supervised fine-tuning. Using weight ensembling, as described in Section 4.2 and depicted in Figure 2, retains generalization, while still enabling a good adaptation.

A similar effect was found by [37], who reported that when fine-tuning their zero-shot model CLIP on ImageNet, the model generalization decreases while the performance on the adaptation domain drastically increases. We argue that such a phenomenon is likely to occur to any model that is fine-tuned on a less diverse dataset compared to the initial training dataset. (In case of CLIP, the initial training dataset consists of 400 million images which is approximately 312 times bigger than ImageNet).

Figure 2. Difference of error for a moderate and a stronger adaptation, corresponding to a learning rate of  $10^{-4}$  and  $10^{-3}$ , respectively. An ImageNet pre-trained ResNet-50 is adapted on one of the corruptions from ImageNet-C at severity 3 and evaluated on all corruptions and the source domain.

Figure 3. Average online classification error rate (%) over 5 runs for the task of *continual* TTA on the highest corruption severity level 5 of ImageNet-C. Varying sequence lengths are considered for a ResNet-50.

**Stability** Undoubtedly, the most critical aspect for a successful universal TTA is stability. Although TENT [51] has demonstrated a successful adaptation to a single domain shift at a time, we empirically show in Appendix A.2 that its performance on ImageNet-C degrades to a trivial solution as the length of the test sequence increases. In addition, by considering CIFAR100-C, we also demonstrate that the occurrence of trivial solutions can be triggered when the domain shifts from time to time—a setting which is likely to be encountered in real world applications and denoted as continual TTA by [53]. We further find that an increased domain non-stationarity has an even more severe effect, as the model develops a bias much faster. In Figure 3, we analyze the performance of current state-of-the-art methods in the online continual TTA setting for ImageNet-C, using different numbers of samples per corruption. While all methods successfully reduce the error rate for 5,000 samples per

corruption, only very few methods do not collapse to trivial solutions or again degrade in performance due to the development of a bias when 50,000 samples are considered. We visualize and discuss the latter two aspects in Appendix A.2. These examples clearly demonstrate the necessity of remaining diverse predictions throughout the adaptation.

## 4. Methodology

In this work, we seek to create a method that performs a good, stable, and efficient adaptation across a wide range of different settings and domain shifts while being mostly model agnostic. Before we address the previous findings in more detail, we first establish the basic framework.

To ensure efficiency during test-time, we only update the network’s normalization parameters (BN, GN, and LN) and freeze all others. To improve the stability and adaptation, we exchange the commonly used entropy loss by a certainty and diversity weighted version of the soft likelihood ratio (SLR) loss. The SLR loss [32] has the advantage that its gradients are less dominated by low confidence predictions, which are typically more likely to be incorrect [32]. The weighted soft likelihood ratio loss is then given by

$$\mathcal{L}_{\text{SLR}}(\mathbf{y}_{ti}) = \sum_c w_{ti} y_{tic} \log\left(\frac{y_{tic}}{\sum_{j \in c} y_{tij}}\right); \quad (1)$$

where  $\mathbf{y}_{ti}$  are the softmax probabilities of the network for the  $i$ -th test sample at time step  $t$  and  $w_{ti}$  is its corresponding weight. Since the SLR loss encourages to scale the networks’ logits larger and larger [32], we propose to clip the softmax probabilities for very high confidence values,



i.e.,  $\mathbf{y}_t \in [0; 0.99]^C$ , where  $C$  is the number of classes. This results in a zero-gradient for probabilities above the clipping value, preventing logit explosion.

To further strengthen the adaptation, we encourage consistency against smaller perturbations. This is achieved by promoting similar outputs between test images which have been identified as certain and diverse ( $\mathbf{x}_t^d$ ) and an augmented view of them. We use color jitter, affine transformations, and horizontal flipping to generate the augmented view  $\mathbf{x}_t^a = \text{Aug}(\mathbf{x}_t^d)$  with predictions  $\mathbf{y}_t^a$ . Subsequently, a weighted consistency loss based on the symmetric cross-entropy (SCE) is calculated

$$L_{\text{SCE}}(\mathbf{y}_{ti}^d; \mathbf{y}_{ti}^a) = \frac{w_{ti}^d}{2} \sum_{c=1}^C y_{tic}^d \log y_{tic}^a + \sum_{c=1}^C y_{tic}^a \log y_{tic}^d : \quad (2)$$

We leverage the SCE loss due to its tolerance towards label noise [54], which is especially important in the setting of self-training where pseudo-labels can be inaccurate.

#### 4.1. Certainty and diversity weighting

Our analysis in Section 3 and Appendix A.2 suggests that it is essential to prevent the model from becoming biased or, worse, collapse to a trivial solution during test-time. Therefore, we introduce a diversity criterion, similar to [33], which ensures that diverse samples are favored in comparison to samples that are similar to the central tendency of recent model predictions. Unlike [33], we propose a diversity weighting that does not require dataset-specific hyperparameters. We begin by tracking the recent tendency of a model’s prediction with an exponential moving average  $\mathbf{y}_{t+1} = \mathbf{y}_t + \frac{(1 - \beta)}{N_b} \sum_{i=1}^{N_b} \mathbf{y}_{ti}$ , setting  $\beta = 0.9$ . To determine a diversity weight for each test sample  $\mathbf{x}_{ti}$ , the cosine similarity between the current model output  $\mathbf{y}_t$  and the tendency of the recent outputs  $\mathbf{y}_t$  is computed as follows

$$w_{\text{div};ti} = \frac{\mathbf{y}_{ti}^\top \mathbf{y}_t}{\|\mathbf{y}_{ti}\| \|\mathbf{y}_t\|} : \quad (3)$$

This strategy has the advantage that if the model output is uniform, uncertain predictions receive a smaller weight, which prevents the incorporation of errors into the model. However, if the model output is biased towards some classes, uncertain predictions will have a large weight, thus promoting error accumulation. Therefore, we additionally utilize certainty weighting based on the negative entropy  $w_{\text{cert};ti} = \frac{1}{H(\mathbf{y}_{ti})} = \frac{1}{-\sum_{c=1}^C y_{tic} \log y_{tic}}$ . To remove model and data dependencies, such as the model’s calibration or the number of classes, we normalize the certainty and diversity weights to be in unit range. To pull apart non-reliable and non-diverse samples from reliable and diverse ones, we take the exponential of the product of diversity and certainty weights, scaled by a temperature :

$$w_t = \exp \frac{w_{\text{div};t} w_{\text{cert};t}}{\tau} : \quad (4)$$

To re-emphasize diversity, all weights of samples whose diversity is less than the mean diversity are set to zero, i.e.,  $w_{ti} = 0$  if  $w_{\text{div};ti} < \text{mean}(w_{\text{div};t})$ .

#### 4.2. Weight ensembling

Since our analysis in Section 3 revealed that self-training is likely to cause a loss of generalization and catastrophic forgetting, we propose weight ensembling. It averages the weights of the source model which potentially has good generalization capabilities and the adapted model, which typically better models the current distribution. Previous literature supports that weight-averaging two models works, if they remain in the same basin of the loss landscape [7]. This is usually true for models which are fine-tuned from the same pre-trained checkpoint [7, 16, 56]. Specifically, we continually ensemble the weights of the initial source model  $\theta_0$  and the weights of the current model  $\theta_t$  at time step  $t$  using an exponential moving average of the form

$$\theta_{t+1} = \beta \theta_t + (1 - \beta) \theta_0 : \quad (5)$$

where  $\beta$  is a momentum term, balancing adaptation and generalization. Since we only update normalization parameters, the memory overhead for storing source weights is neglectable. The advantages of equipping TENT with our weight ensembling approach, using a momentum term five times larger as the learning rate, are illustrated in Figure 2. Clearly, the strategy prevents drastic decreases in performance on unseen domains while still allowing good adaptation. By inspecting the last column, it also becomes apparent that catastrophic forgetting is largely mitigated.

#### 4.3. Prior correction during test-time

Consider the scenario where no domain shift exists and only the class distributions between the training and test data differ. In this case, a non-adapted model will underperform because the learned posterior  $q(y|x)$  will deviate from the actual posterior  $p(y|x)$  due to the shift in priors, i.e.,  $q(y) \neq p(y)$ . However, as shown by [38], optimal performance can be recovered by correcting the deviation in posterior according to  $p(y|x) = q(y|x) \frac{p(y)}{q(y)}$ . In the context of online TTA with temporally correlated and thus highly imbalanced data, such performance degradation can easily occur. For example, when the actual class prior is highly dynamic. Since our diversity weighting aims to stabilize model adaptation by preventing the network from learning any biases, there will be a discrepancy between the class priors. Therefore, we propose a prior correction that reweights the final predictions by  $\frac{p(y)}{q(y)}$  without influencing the adaptation.

As a result of diversity weighting, we assume a uniform distribution for the learned prior  $q(y)$ . To determine the actual class prior  $p(y)$ , we suggest to use the sample mean over the current softmax predictions  $\mathbf{y}_{ti}$  as a proxy  $\hat{p}_t =$

$\frac{1}{N_b} \sum_{i=1}^{N_b} \mathbf{y}_{ti}$ . Since only  $N_b$  test samples are considered for the estimation of the actual class prior, the resulting estimate will be inaccurate. Therefore, an adaptive additive smoothing scheme is proposed

$$\mathbf{p}_t = \frac{\hat{\mathbf{p}}_t + \mathbf{1}}{1 + N_c}, \quad (6)$$

where  $N_c$  denotes the number of classes and  $\alpha$  is an adaptive smoothing factor that is determined by the ratio  $\alpha = \max(1 - N_b, 1 - N_c) = \max_c \hat{\mathbf{p}}_{tc}$ . The idea behind this ratio is that if the class distribution within a batch tends to be uniform,  $\alpha \rightarrow 1$ , a strong smoothing is applied ensuring that no class is favored. If the class distribution is strongly biased towards one class,  $\alpha \rightarrow 0$ , minor smoothing is applied. In settings with highly imbalanced data, weighting the network’s outputs with a smoothed estimate of the class prior can significantly improve the predictions. Uncertain data points can be corrected by taking class prior information into account, while not degrading performance when a uniform class distribution is present.

## 5. Experiments

**Datasets** We evaluate our approach for a wide range of different domain shifts, including corruptions and natural shifts. Following [51], we consider the corruption benchmark [13] consisting of CIFAR10-C, CIFAR100-C, and ImageNet-C. These datasets include 15 types of corruptions with 5 severity levels applied to the validation and test images of ImageNet (IN) and CIFAR, respectively [19]. For the natural domain shifts, we consider ImageNet-R [12], ImageNet-Sketch [52], as well as a variation of ImageNet-D [39], which we denote as ImageNet-D109. While ImageNet-R contains 30,000 examples depicting different renditions of 200 IN classes, ImageNet-Sketch contains 50 sketches for each of the 1,000 IN classes. ImageNet-D is based on DomainNet [35], which contains 6 domain shifts (clipart, infograph, painting, quickdraw, real, sketch), and considers samples that are one of the 164 classes that overlap with ImageNet. For ImageNet-D109, we use all classes that have a one-to-one mapping from DomainNet to ImageNet, resulting in 109 classes. We omit the domain *quickdraw* in our experiments since many examples cannot be attributed to a class [40].

**Considered settings** All experiments are performed in the online TTA setting, where the predictions are evaluated immediately. To assess the performance of each method for universal TTA, we consider four different settings. The first is the *continual* benchmark [53], where the model is adapted to a sequence of  $K$  different domains  $D$  without knowing when a domain shift occurs, i.e.  $[D_1; D_2; \dots; D_K]$ . For the corruption datasets, the domain sequence comprises 15 corruptions, each encountered at the highest severity level 5. For ImageNet-R and ImageNet-Sketch there exists only a single

domain and for ImageNet-D109 the domains are encountered in alphabetical order. The second setting is denoted as *mixed domains*. Since in this case the test data of all domains are randomly shuffled before the adaptation, consecutive test samples are likely to originate from different domains. Third, we examine a *correlated* setting which is similar to the continual one, since the domains are also encountered sequentially. However, in the correlated setting, the data of each domain is sorted by the class label rather than randomly shuffled, resulting in class imbalanced batches. Finally, we also consider the situation where the domains are mixed and the sequence is temporally correlated. Single domain settings are not explicitly considered since any method that succeeds in the continual setting, will also succeed in the single domain setting.

**Implementation details** Following previous work [53], a pre-trained WideResNet-28 (WRN-28) [61] and ResNeXt-29 [58] is used for CIFAR10-to-CIFAR10-C and CIFAR100-to-CIFAR100-C, respectively. For the ImageNet datasets a source pre-trained ResNet-50, a VisionTransformer [6] in its base version with an input patch size of 16 × 16 (Vit-b-16), and a SwinTransformer [23] in its base version (Swin-b) are used. Note that for our method, we additionally ablate 28 pre-trained networks available in PyTorch in Appendix B.1. We follow the implementation of [51], using the same hyperparameters. Further, we fix the momentum term  $\mu$  used for weight ensembling to 0.99 and set the temperature  $\tau$  to  $\frac{1}{3}$ .

**Baselines** We compare our approach to other source-free TTA methods that also use an arbitrary off-the-shelf pre-trained model. In particular, we compare to TENT non-episodic [51], EATA [33], SAR [34], CoTTA [53], RoTTA [60], AdaContrast [4], RMT [5], and LAME [2]. In addition, we consider the non-adapted model (source) and the normalization-based method BN-1, which recalculates the batch normalization statistics using the current test batch. As metric, we use the error rate.

### 5.1. Results

**Results for continual TTA** Table 1 shows the results for online continual TTA, with results worse than the source performance highlighted in red. We find that LAME significantly decreases the performance on all continual benchmarks, due to its tendency of predicting only a reduced number of classes in each batch. This can also be seen in Figure 7 in the appendix. While SAR is able to adapt to corrupted data for all architectures, its adaptation capabilities for natural domain shifts are limited when using transformers. Further, although SAR proposed a model restore approach to avoid performance degradation, the approach lacks generalization. The effectiveness of TENT also heavily depends on the domain shift and architecture, as Vit-b-16 provides clear benefits for IN-C and IN-R, but fails for IN-D109, for example. However, by equipping TENT with a diversity criterion,

Table 1. Average online classification error rate (%) over 5 runs in the *continual* TTA setting.

| Dataset    | Architecture | Source | BN-1 | TENT | EATA | SAR  | CoTTA | RoTTA | AdaCont. | RMT  | LAME | ROID (ours) |
|------------|--------------|--------|------|------|------|------|-------|-------|----------|------|------|-------------|
| CIFAR10-C  | WRN-28       | 43.5   | 20.4 | 20.0 | 17.9 | 20.4 | 16.5  | 19.3  | 18.5     | 17.0 | 64.3 | 16.2 0.05   |
| CIFAR100-C | ResNext-29   | 46.4   | 35.4 | 62.2 | 32.2 | 32.0 | 32.8  | 34.8  | 33.5     | 30.2 | 98.5 | 29.3 0.04   |
| IN-C       | ResNet-50    | 82.0   | 68.6 | 62.6 | 58.0 | 61.9 | 63.1  | 67.3  | 65.5     | 59.9 | 93.5 | 54.5 0.1    |
|            | Swin-b       | 64.0   | 64.0 | 64.0 | 52.8 | 63.7 | 59.3  | 62.7  | 58.1     | 52.6 | 84.8 | 47.0 0.26   |
|            | ViT-b-16     | 60.2   | 60.2 | 54.5 | 49.8 | 51.7 | 77.0  | 58.3  | 57.0     | 72.9 | 79.9 | 45.0 0.09   |
| IN-R       | ResNet-50    | 63.8   | 60.5 | 57.6 | 54.2 | 57.5 | 57.4  | 60.7  | 58.9     | 56.1 | 99.3 | 51.2 0.11   |
|            | Swin-b       | 54.2   | -    | 53.8 | 49.9 | 53.0 | 52.9  | 53.0  | 52.3     | 47.4 | 92.7 | 45.8 0.12   |
|            | ViT-b-16     | 56.0   | -    | 53.3 | 49.0 | 48.6 | 69.6  | 54.4  | 54.2     | 68.8 | 95.2 | 44.2 0.13   |
| IN-Sketch  | ResNet-50    | 75.9   | 73.6 | 69.5 | 64.5 | 68.4 | 69.5  | 70.8  | 73.0     | 68.4 | 99.8 | 64.3 0.16   |
|            | Swin-b       | 68.4   | -    | 68.7 | 60.5 | 72.6 | 71.0  | 67.1  | 64.4     | 69.0 | 94.6 | 58.8 0.15   |
|            | ViT-b-16     | 70.6   | -    | 70.5 | 59.7 | 70.6 | 95.5  | 69.0  | 68.3     | 86.8 | 99.5 | 58.6 0.07   |
| IN-D109    | ResNet-50    | 58.8   | 55.1 | 52.9 | 51.6 | 52.2 | 50.8  | 52.3  | 50.4     | 49.4 | 85.0 | 48.0 0.06   |
|            | Swin-b       | 51.4   | -    | 66.1 | 47.5 | 54.2 | 49.9  | 48.7  | 47.3     | 47.6 | 86.3 | 45.1 0.10   |
|            | ViT-b-16     | 53.6   | -    | 84.0 | 47.4 | 57.4 | 73.4  | 51.2  | 49.7     | 74.2 | 88.0 | 45.0 0.04   |

Table 2. Average online classification error rate (%) over 5 runs in the *mixed domains* TTA setting.

| Dataset    | Architecture | Source | BN-1 | TENT | EATA | SAR  | CoTTA | RoTTA | AdaCont. | RMT  | LAME | ROID (ours) |
|------------|--------------|--------|------|------|------|------|-------|-------|----------|------|------|-------------|
| CIFAR10-C  | WRN-28       | 43.5   | 33.8 | 44.1 | 28.6 | 33.8 | 32.5  | 33.4  | 26.2     | 31.0 | 75.2 | 28.0 0.12   |
| CIFAR100-C | ResNext-29   | 46.4   | 45.8 | 82.5 | 36.9 | 45.5 | 43.1  | 45.4  | 41.8     | 38.6 | 98.4 | 35.0 0.04   |
| IN-C       | ResNet-50    | 82.0   | 82.5 | 86.4 | 72.3 | 79.4 | 76.0  | 78.1  | 90.8     | 75.4 | 95.1 | 69.5 0.13   |
|            | Swin-b       | 64.0   | -    | 62.6 | 56.3 | 60.6 | 63.3  | 62.6  | 66.0     | 55.4 | 64.6 | 55.0 0.26   |
|            | ViT-b-16     | 60.2   | -    | 55.0 | 51.8 | 52.3 | 89.3  | 58.2  | 65.5     | 73.4 | 62.6 | 50.7 0.08   |
| IN-D109    | ResNet-50    | 58.8   | 56.2 | 56.1 | 53.3 | 53.7 | 50.3  | 54.0  | 55.4     | 50.7 | 99.1 | 50.9 0.04   |
|            | Swin-b       | 51.4   | -    | 61.5 | 48.9 | 54.0 | 49.4  | 48.1  | 49.4     | 46.5 | 97.3 | 47.2 0.07   |
|            | ViT-b-16     | 53.6   | -    | 76.7 | 48.6 | 61.4 | 58.0  | 50.5  | 51.4     | 70.8 | 98.8 | 46.9 0.02   |

TENT remains stable in all configurations, suggesting that diversity also contributes to become more model and shift agnostic. This might also be the reason, why methods like EATA, AdaContrast and RoTTA remain stable, as each of them either explicitly enforce diversity or leverages a diversity buffer. Our method ROID is not only stable, but yields significant performance improvements compared to the second best approach, EATA, which requires dataset specific hyperparameters and access to data from the initial source domain. Note that we additionally verify the effectiveness of ROID for 28 pre-trained networks in Appendix B.1, demonstrating its wide applicability.

**Results for mixed domains** Table 2 illustrates the results for the mixed domains setting. By comparing the performance between the settings *continual* and *mixed domains* for methods such as EATA, SAR, AdaContrast, RMT, and ROID for the transformers, it becomes obvious that adapting to multiple target domains at the same time is more challenging. In case of BN-based architectures, like ResNets, the results can also significantly decrease due to missing improvements of covariate shift mitigation through recalculating the BN statistic. Our method ROID is again not only stable, but performs best or comparable on most benchmarks.

**Results for correlated (+mixed domains)** First, we consider a correlated setting, where samples are sorted by class. Since re-calculating BN statistics now even increases the

Table 3. Average online classification error rate (%) for IN-C (at level 5) and IN-D109 for the *mixed domains correlated* setting, using  $\alpha = 0.01$  and  $\alpha = 0.1$ , respectively.

|          | Method | IN-C      | IN-D109   |
|----------|--------|-----------|-----------|
| Swin-b   | Source | 64.0      | 51.4      |
|          | SAR    | 64.9 0.81 | 53.9 0.52 |
|          | LAME   | 37.4 0.12 | 28.0 0.39 |
|          | ROID   | 28.6 0.16 | 28.3 0.19 |
| ViT-b-16 | Source | 60.2      | 53.6      |
|          | SAR    | 54.3 0.59 | 60.8 0.48 |
|          | LAME   | 36.1 0.15 | 29.2 0.55 |
|          | ROID   | 23.6 0.05 | 29.4 0.13 |

error absolutely by 13.8% to 95.8% for a ResNet-50 on the long ImageNet-C sequence, we only consider transformers based on layer normalization and the same ResNet-26 with group normalization that was used in [62].

The results are presented in Figure 4 (left). Detailed results are further shown in Table 14 and Table 15 in the appendix. Even though SAR was proposed for a correlated setting, in this extreme case of sorted classes and multiple domain shifts, its performance often degrades below the source baseline. A similar trend can also be observed for RoTTA, which also does not show any substantial performance improvements. The only methods that can significantly outperform the source baseline are LAME and ROID. Since LAME tends to predict only a few classes, it performs

Figure 4. Online classification error rate (%) in the *correlated* TTA setting, where samples are sorted by class on the left and for different levels of correlation on the right.

well in the correlated setting, while drastically degrading the performance in previous scenarios. ROID, on the other hand, outperforms LAME on 3 out of 5 datasets, while also showing strong results in other settings. On the right of Figure 4, we illustrate the performance for different degrees of correlation by varying the concentration parameter of a Dirichlet distribution [11, 63]. Prior correction and, consequently, ROID benefit from increasing correlation, as the entropy of the class prior decreases.

Lastly, we investigate the combination of temporally correlated data with mixed domains for IN-C and IN-D109. As shown in Table 3, ROID achieves significantly better and comparable results than existing methods, demonstrating its ability to perform in all scenarios of universal TTA.

**Results for single sample TTA** Updating the model using a single test sample not only yields noisy gradients, but also prevents an accurate estimation of the BN statistics, resulting in a performance degradation. While [5, 25] use a small buffer to store the last  $b$  test samples on the device, this comes with a trade-off between efficiency and accurate BN statistics. To circumvent this issue, we propose to use networks that do not employ BN layers, such as VisionTransformer [6]. These networks allow to recover the batch TTA setting by simply accumulating the gradients of the last  $b$  test samples before updating the model. As shown in Table 9, this provides the same results as before, with no computational overhead and significantly reduced memory requirements.

## 5.2. Ablation studies

In Appendix B, we further analyze the efficiency, catastrophic forgetting, and the momentum used for weight ensembling. We find that ROID successfully maintains its knowledge about the initial training domain while being computationally efficient.

**Component analysis** In Table 4, we analyze the components of ROID. In general, the component analysis underscores our primary hypotheses and findings. Certainty and diversity based loss weighting helps in all scenarios by miti-

Table 4. Average online classification error rate (%) over 5 runs for different configurations and settings.

| Method              | <i>continual</i> | <i>mixed</i> | <i>correlated</i> | <i>mix. + corr.</i> |
|---------------------|------------------|--------------|-------------------|---------------------|
| Source              | 61.7             | 57.7         | 54.6              | 48.8                |
| SLR                 | 52.6             | 66.7         | 80.0              | 88.1                |
| + Loss weighting    | 46.1             | 46.4         | 60.4              | 61.1                |
| + Weight ensembling | 45.0             | 46.9         | 46.7              | 44.9                |
| + Consistency       | 43.9             | 46.0         | 45.7              | 43.8                |
| + Prior correction  | 43.9             | 45.9         | 26.8              | 23.5                |

gating the development of a model bias. Weight ensembling demonstrates its effectiveness in settings where the model has to adapt sequentially to multiple narrow distributions, such as in the *continual* and *correlated* setting. It does not contribute, when a broad distribution is present (*mixed domains*). For the difficult adaptation in correlated settings, weight ensembling also serves as a corrective measure. It addresses suboptimal adaptations over time by continually incorporating a small percentage of the source weights. Finally, prior correction shows its strong suits in *correlated* settings and upholds performance when a uniform class distribution is present. Further details and discussions are located in B.6.

## 6. Conclusion

In this work, we derive all practically relevant settings and denote this as *universal TTA*. By further highlighting several challenges which can arise when conducting self-training during test-time, namely the loss of generalization, model bias, and trivial solutions, we introduce a new TTA method: ROID. To retain generalization, ROID continually weight-averages the source and adapted model. For promoting stability and encourage diverse predictions, a certainty and diversity weighted SLR loss is used. To compensate for prior shifts that can occur during test-time, a novel adaptive prior correction scheme is proposed. We set new standards in the field of online universal TTA.



## References

- [1] 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. [3](#)
- [2] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022. [1](#), [3](#), [6](#)
- [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. [3](#), [6](#)
- [5] Mario Döbler, Robert A Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7704–7714, 2023. [3](#), [6](#), [8](#), [13](#)
- [6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. [6](#), [8](#)
- [7] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020. [5](#)
- [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. [3](#)
- [11] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [1](#), [3](#), [8](#)
- [12] 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](#), [6](#)
- [13] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. [3](#), [6](#)
- [14] 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](#)
- [15] 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](#)
- [16] Pavel Izmailov, Dmitrii Podoprikin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018. [5](#)
- [17] 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](#)
- [18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. [3](#)
- [19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. [3](#), [6](#)
- [20] 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. [3](#)
- [21] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. *Advances in Neural Information Processing Systems*, 34:22968–22981, 2021. [2](#), [13](#)
- [22] 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. [3](#)
- [23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. [6](#)
- [24] 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](#)
- [25] Robert A Marsden, Mario Döbler, and Bin Yang. Introducing intermediate domains for effective self-training during test-time. *arXiv preprint arXiv:2208.07736*, 2022. [1](#), [3](#), [8](#)
- [26] 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
- [27] 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. 3, 18
- [28] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. *arXiv preprint arXiv:2008.12197*, 2020. 2
- [29] 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
- [30] 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. 3
- [31] 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
- [32] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambaich, 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. 3, 4
- [33] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 3, 5, 6, 18
- [34] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 2, 3, 6, 13
- [35] 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. 6
- [36] Joaquin Quiñero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 1
- [37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3
- [38] Amélie Royer and Christoph H Lampert. Classifier adaptation at prediction time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1401–1409, 2015. 5
- [39] Evgenia Rusak, Steffen Schneider, Peter Vincent Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Imagenet-d: A new challenging robustness dataset inspired by domain adaptation. In *ICML 2022 Shift Happens Workshop*, 2022. 6
- [40] 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. 6
- [41] 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
- [42] 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. 2
- [43] 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
- [44] 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, 3
- [45] 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
- [46] 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
- [47] Wilhelm Truheden, 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
- [48] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Bochoon, 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
- [49] 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
- [50] 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

- [51] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. [1](#), [3](#), [4](#), [6](#), [13](#)
- [52] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. [6](#)
- [53] 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](#), [3](#), [4](#), [6](#), [13](#), [18](#)
- [54] 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. [5](#)
- [55] 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](#)
- [56] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. [5](#)
- [57] 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](#)
- [58] 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. [6](#)
- [59] 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](#)
- [60] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023. [1](#), [2](#), [3](#), [6](#)
- [61] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. [6](#)
- [62] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021. [3](#), [7](#)
- [63] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021. [8](#)
- [64] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Pro-*

*ceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. [2](#)