

Continual Test-Time Domain Adaptation

Qin Wang¹ Olga Fink^{1,3*} Luc Van Gool^{1,4} Dengxin Dai²

¹ETH Zurich, Switzerland ²MPI for Informatics, Germany ³EPFL, Switzerland ⁴KU Lueven, Belgium
{qin.wang, vangool, dai}@vision.ee.ethz.ch olga.fink@epfl.ch

Abstract

Test-time domain adaptation aims to adapt a source pre-trained model to a target domain without using any source data. Existing works mainly consider the case where the target domain is static. However, real-world machine perception systems are running in non-stationary and continually changing environments where the target domain distribution can change over time. Existing methods, which are mostly based on self-training and entropy regularization, can suffer from these non-stationary environments. Due to the distribution shift over time in the target domain, pseudo-labels become unreliable. The noisy pseudo-labels can further lead to error accumulation and catastrophic forgetting. To tackle these issues, we propose a continual test-time adaptation approach (CoTTA) which comprises two parts. Firstly, we propose to reduce the error accumulation by using weight-averaged and augmentation-averaged predictions which are often more accurate. On the other hand, to avoid catastrophic forgetting, we propose to stochastically restore a small part of the neurons to the source pre-trained weights during each iteration to help preserve source knowledge in the long-term. The proposed method enables the long-term adaptation for all parameters in the network. CoTTA is easy to implement and can be readily incorporated in off-the-shelf pre-trained models. We demonstrate the effectiveness of our approach on four classification tasks and a segmentation task for continual test-time adaptation, on which we outperform existing methods. Our code is available at <https://qin.ee/cotta>.

1. Introduction

Test-time domain adaptation aims to adapt a source pre-trained model by learning from the unlabeled test (target) data during inference time. Due to the domain shift between source training data and target test data, an adaptation is necessary to achieve good performance. For example, a semantic segmentation model trained on data from

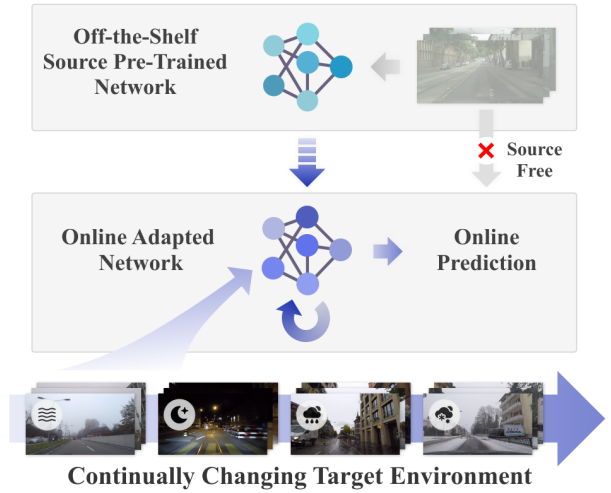


Figure 1. We consider the online continual test-time adaptation scenario. The target data is provided in a sequence and from a continually changing environment. An off-the-shelf source pre-trained network is used to initialize the target network. The model is updated online based on the current target data, and the predictions are given in an online fashion. The adaptation of the target network does not rely on any source data. Existing methods often suffer from error accumulation and forgetting which result in performance deterioration over time. Our method enables long-term test-time adaptation under continually changing environments.

clear weather conditions can suffer significant performance deterioration when tested on snowy night conditions [50]. Similarly, a pre-trained image classification model can also suffer this phenomenon when tested on corrupted images resulting from sensor degradation. Due to privacy concerns or legal constraints, the source data is generally considered unavailable during inference time under this setup, making it a more challenging but more realistic problem than unsupervised domain adaptation. In many scenarios, the adaptation also needs to be performed in an online fashion. Therefore, test-time adaptation is critical to the success of real-world machine perception applications under domain shift.

Existing works on test-time adaptation often tackle the distribution shift between the source domain and a fixed target domain by updating model parameters using pseudo-labels or entropy regularization [43, 61]. These self-training

*The corresponding author

methods have been proven to be effective when the test data are drawn from the same stationary domain. However, they can be unstable [48] when the target test data originates from an environment which is continually changing. There are two aspects that contribute to this: Firstly, under the continually changing environment, the pseudo-labels become noisier and mis-calibrated [13] because of the distribution shift. Therefore, early prediction mistakes are more likely to result in error accumulation [4]. Secondly, as the model is being continually adapted to new distributions for a long time, knowledge from the source domain is harder to preserve, leading to catastrophic forgetting [11, 41, 45].

Aiming to tackle these problems under the continually changing environment, this work focuses on the practical problem of online continual test-time adaptation. As shown in Figure 1, the goal is to start from an off-the-shelf source pre-trained model, and continually adapt it to the current test data. Under this setup, we assume that the target test data is streamed from a continually changing environment. The prediction and updates are performed online, meaning that the model will only have access to the current stream of data without having access to the full test data nor any source data. The proposed setup is very relevant for real-world machine perception systems. For example, surrounding environments are continually changing for autonomous driving systems (e.g. weather change from sunny to cloudy then to rainy). They can even change abruptly (e.g. when a car exits a tunnel and the camera gets suddenly over-exposed). A perception model need to adapt itself and make decisions online under these non-stationary domain shifts.

To effectively adapt the pre-trained source model to the continually changing test data, we propose a continual test-time adaptation approach (CoTTA) which tackles the two main limitations of existing methods. The first component of the proposed method aims to alleviate error accumulation. We propose to improve the pseudo-label quality under the self-training framework in two different ways. On the one hand, motivated by the fact that the mean teacher predictions often have a higher quality than the standard model [55], we use a weight-averaged teacher model to provide more accurate predictions. On the other hand, for test data which suffers larger domain gap, we use the augmentation-averaged predictions to further boost the quality of pseudo-labels. The second component of the proposed method aims to help preserve the source knowledge and avoid forgetting. We propose to stochastically restore a small part of neurons in the network back to the pre-trained source model. By reducing error accumulation and preserving knowledge, CoTTA enables long-term adaptation in a continuously changing environment, and makes it possible to train all parameters of the network. In contrast, previous methods [43, 61] can only train batchnorm parameters.

It is worth pointing out that our approach can be eas-

ily implemented. The weight-and-augmentation-averaged strategy and the stochastic restoration can be readily incorporated into any off-the-shelf pre-trained model without the need to re-train it on source data. We demonstrate the effectiveness of our proposed approach on four classification tasks and a segmentation task for continual test-time adaptation, on which we significantly improve performance over existing methods. Our contributions are summarized below:

- We propose a continual test-time adaptation approach which can effectively adapt off-the-shelf source pre-trained models to continually changing target data.
- Specifically, we reduce the error accumulation by using weight-averaged and augmentation-averaged pseudo-labels that are more accurate.
- The long-term forgetting effect is alleviated by explicitly preserving the knowledge from the source model.
- The proposed approach significantly improves the continual test-time adaptation performance on both classification and segmentation benchmarks.

2. Related Work

2.1. Domain Adaptation

Unsupervised domain adaptation (UDA) [44, 46] aims to improve the target model performance in the presence of a domain shift between the labeled source domain and unlabeled target domain. During training, UDA methods often align the feature distributions between the two domains using discrepancy losses [39] or adversarial training [12, 58]. Alternatively, the alignment can also be done in the input space [18, 67]. In recent years, self-training has also shown promising results by iteratively using gradually-improving target pseudo-labels to train the network [19, 36, 62, 75].

2.2. Test-time Adaptation

Test-time adaptation is also referred to as source-free domain adaptation in some references [28, 66]. Unlike domain adaptation which requires access to both source and target data for adaptation, test-time adaptation methods do not require any data from the source domain for adaptation. Some existing works [29, 33, 68] utilize generative models to support the feature alignment in absence of source data.

Another popular direction is to finetune the source model without explicitly conducting domain alignment. Test entropy minimization (TENT) [61] takes a pre-trained model and adapts to the test data by updating the trainable parameters in Batchnorm layers using entropy minimization. Source hypothesis transfer (SHOT) [37] utilizes both entropy minimization and a diversity regularizer for the adaptation. SHOT requires using source data to train a specialized source model using the label-smoothing technique

Table 1. The difference between our proposed continual test-time adaptation and related adaptation settings.

Setting	Data		Learning	
	Source	Target	Train stage	Test stage
standard domain adaptation	Yes	stationary	Yes	No
standard test-time training [54]	Yes	stationary	Yes (aux task)	Yes
fully test-time adaptation [61]	No	stationary	No (pre-trained)	Yes
continual test-time adaptation	No	continually changing	No (pre-trained)	Yes

with the weight normalization layer. Thus, it cannot support the use of an arbitrary pre-trained model. [43] proposes to apply a diversity regularizer combined with an input transformation module to further improve the performance. [23] uses a separate normalization convolutional network to normalize test images from new domains. [22] only updates the final classification layer during inference time using pseudo-prototypes. [74] analyzes the problem in a Bayesian perspective and proposes a regularized entropy minimization procedure at test-time adaptation, which requires approximating density during training time. Updating the statistics in the Batch Normalization layer using the target data is a different path which also shows promising results [21, 34, 70]. While most existing works focus on image classification, [20, 27, 38] extend test-time adaptation to semantic segmentation. Standard test-time adaptation considers the offline scenario where access to the full set of test data is provided for the training. This is often unrealistic for online machine perception applications. Most existing works (except TENT variants [60]) also require the re-training of the source model to support the test-time adaptation. Therefore, they cannot directly use off-the-shelf pre-trained model from the source domain.

2.3. Continuous Domain Adaptation

Unlike standard domain adaptation which assumes a specific target domain, continuous domain adaptation considers the adaptation problem with **continually changing target data**. Continuous Manifold Adaptation (CMA) [17] is an early work which considers adaptation to evolving domains. Incremental adversarial domain adaptation (IADA) [63] adapts to continually changing domains by adversarially aligning source and target features. [59] aims to continually adapt the unseen visual domain while alleviate the forgetting on the seen domain without retaining the source training data. [3] aims to adapt to gradually changing domains by making use of the assumption of continuity between gradually varying domains. Existing continuous domain adaptation methods need to have access to data from both the source and target domains in order to align the distributions.

The main focus of this paper is **continual test-time adaptation**, which additionally considers the adaptation at test-time without accessing the source data. While this is a realistic scenario for machine perception systems in the real world, there are very limited number of approaches which are applicable to such scenarios. In theory, the online ver-

sion of TENT [61] could adapt under this setup by continually updating the BN parameters using the entropy loss. However, it can suffer from error accumulation because of mis-calibrated predictions. Test-time training (TTT) [54] could also continually update the feature extractor using supervision from the rotation prediction auxiliary task. However, it requires re-training of the source model using the source data to learn the auxiliary task. Therefore, it cannot be considered as source-free for the full pipeline and does not support off-the-shelf source pre-trained models.

2.4. Continual Learning

Continual learning [10] and lifelong learning [45] are closely related to the continuous adaptation problems as a potential cure to the catastrophic forgetting. Continual learning methods can often be categorized into replay-based [49] and regularization-based [53, 72] methods. The latter can further be divided into data-focused methods, such as learning without forgetting (LwF) [35], and prior-focused methods, such as elastic weight consolidation (EWC) [24]. Ideas from continual learning are adopted for continuous domain adaptation approaches [3, 30].

2.5. Domain Generalization

This work is also related to domain generalization [42] in a broad sense, because of the shared goal of improving performance on potentially changing target domains. A number of works have also shown that data augmentation [52] during training [14, 16, 32, 69] and during testing [1, 40, 73] can improve model robustness and generalizability. Domain randomization is one of the most popular methods which improves the model generalizability by learning from different synthesis parameters of simulation environments [56, 57]. Unlike domain generalization methods which mostly aim to train a more generalizable neural network from the source domain, this work focuses on improving the performance of existing pre-trained neural networks during test-time by using the unlabeled online data from the continually changing target domain.

3. Continual Test-Time Domain Adaptation

3.1. Problem Definition

Given an existing pre-trained model $f_{\theta_0}(x)$ with parameters θ trained on the source data $(\mathcal{X}^S, \mathcal{Y}^S)$, we aim at improving the performance of this existing model during in-

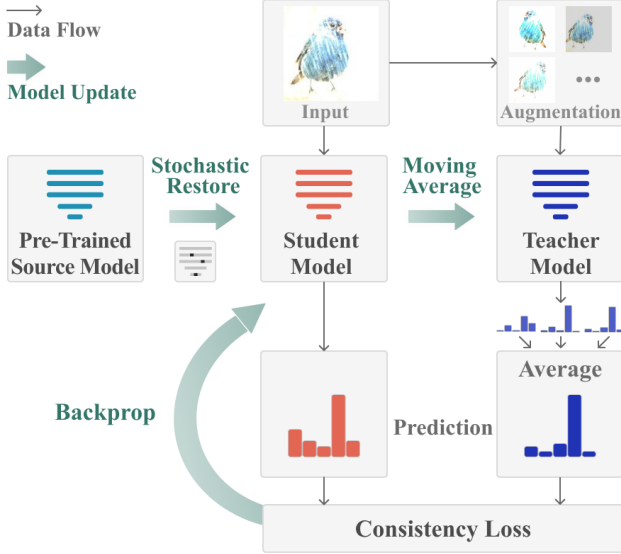


Figure 2. An overview of the proposed continual test-time adaptation (CoTTA) approach. CoTTA adapts from an off-the-shelf source pre-trained network. Error accumulation is mitigated by using a teacher model to provide weight-averaged pseudo-labels and using multiple augmentations to average the predictions. Knowledge from the source data is preserved by stochastically restoring a small number of elements of trainable weights.

ference time for a continually changing target domain in an online fashion without having access to any source data. Unlabeled target domain data \mathcal{X}^T is provided sequentially and the model only have access to the data of the current time step. At time step t , target data x_t^T is provided as input and the model f_{θ_t} needs to make the prediction $f_{\theta_t}(x_t^T)$ and adapts itself accordingly for future inputs $\theta_t \rightarrow \theta_{t+1}$. The data distribution of x_t^T is continually changing. The model is evaluated based on the online predictions.

This setup is largely motivated by the need of machine perception applications in continually changing environments. For example, the surrounding environment is continually changing for autonomous driving cars because of location, weather, and time. Perception decisions need to be made online and models need to be adapted.

We list the main differences between our online continual test-time adaptation setup with existing adaptation setups in Table 1. Compared to previous setups which focus on a fixed target domain, we consider the long-term adaptation on continually changing target environments.

3.2. Methodology

We propose an adaptation method for the online continual test-time adaptation setup. The proposed method takes an off-the-shelf source pre-trained model and adapts it to the continually changing target data in an online fashion. Motivated by the fact that error accumulation is one of the key bottlenecks in the self-training framework, we propose to use weight-and-augmentation-averaged pseudo-labels to

reduce error accumulation. In addition, to help reduce forgetting in continual adaptation, we propose to explicitly preserve information from the source model. An overview of the proposed method is presented in Figure 2.

Source Model Existing works on test-time adaptation often require special treatment in the training process of the source model to improve domain generalization ability and to facilitate the adaptation. For example, during source training, TTT [54] has an additional auxiliary rotation prediction branch to train to facilitate the target adaptation supervision. This requires a retraining on the source data, and makes it impossible to reuse existing pre-trained models. In our proposed test-time adaptation method, we lift this burden and do not require a modification of the architecture or an additional source training process. Therefore, any existing pre-trained models can be used without retraining on the source. We will show in the experiments that our method can work on a wide range of pre-trained networks including ResNet variants and Transformer-based architectures.

Weight-Averaged Pseudo-Labels Given target data x_t^T and the model f_{θ_t} , the common test-time objective under the self-training framework is to minimize the cross-entropy consistency between the prediction $\hat{y}_t^T = f_{\theta_t}(x_t^T)$ and a pseudo-label. For example, directly using the model prediction itself as the pseudo-label leads to the training objective of TENT [61] (i.e. entropy minimization). While this works for a stationary target domain, the quality of pseudo-labels can drop significantly for continually changing target data because of the distribution shift.

Motivated by the observation that weight-averaged models over training steps often provide a more accurate model than the final model [47, 55], we use a weight-averaged teacher model $f_{\theta'}$ to generate the pseudo-labels. At time-step $t = 0$, the teacher network is initialized to be the same as the source pre-trained network. At time-step t , the pseudo-label is first generated by the teacher $\hat{y}_{tc}^T = f_{\theta'}(x_t^T)$. The student f_{θ_t} is then updated by the cross-entropy loss between the student and teacher predictions:

$$\mathcal{L}_{\theta_t}(x_t^T) = - \sum_c \hat{y}_{tc}^T \log \hat{y}_{tc}^T, \quad (1)$$

where \hat{y}_{tc}^T is the probability of class c in the teacher model's soft pseudo-label prediction, and \hat{y}_{tc}^T is the prediction from the main model (student). The loss enforces a consistency between the teacher and student predictions.

After the update of the student model $\theta_t \rightarrow \theta_{t+1}$ using Equation 1, we update the weights of the teacher model by exponential moving average using the student weights:

$$\theta'_{t+1} = \alpha \theta'_t + (1 - \alpha) \theta_{t+1}, \quad (2)$$

where α is a smoothing factor. Our final prediction for the input data x_t^T is the class with the highest probability in \hat{y}_t^T .

The benefits of the weight-averaged consistency are two-fold. On the one hand, by using the often more accurate [47] weight-averaged prediction as the pseudo-label target, our model suffers less from the error accumulation during the continual adaptation. On the other hand, the mean teacher prediction \hat{y}_t^T encodes the information from models in past iterations and is, therefore, less likely to suffer from catastrophic forgetting in long-term continual adaptation and improve the generalization capability to new unseen domains. This is inspired by the mean teacher method proposed in [55] in semi-supervised learning.

Augmentation-Averaged Pseudo-Labels Data augmentation during training time [52] has been widely applied to improve model performance. Different augmentation policies are often manually designed [26] or searched [9] for different datasets. While test-time augmentation has also been proven to be able to improve robustness [5, 54], the augmentation policies are generally determined and fixed for a specific dataset without considering the distribution change during inference time. Under a continually changing environment, test distributions can change dramatically, which may make the augmentation policy invalid. Here, we take the test-time domain shift into account and approximate the domain difference by prediction confidence. The augmentation is only applied when the domain difference is large, to reduce error accumulation.

$$\tilde{y}_t^T = \frac{1}{N} \sum_{i=0}^{N-1} f_{\theta'_i}(x_t^T), \quad (3)$$

$$y_t^T = \begin{cases} \hat{y}_t^T, & \text{if } \text{conf}(f_{\theta_0}(x_t^T)) \geq p_{th} \\ \tilde{y}_t^T, & \text{otherwise,} \end{cases} \quad (4)$$

where \tilde{y}_t^T is the augmentation-averaged prediction from the teacher model, \hat{y}_t^T is the direct prediction from the teacher model, $\text{conf}(f_{\theta_0}(x_t^T))$ is the source pre-trained model's prediction confidence on the current input x_t^T , and p_{th} is a confidence threshold. By calculating the prediction confidence on the current input x_t^T using the pre-trained model f_{θ_0} in Equation 4, we attempt to approximate the domain difference between the source and the current domain. We hypothesize that a lower confidence indicates a larger domain gap and a relatively high confidence level indicates a smaller domain gap. Therefore, when the confidence is high and larger than the threshold, we directly use \hat{y}_t^T as our pseudo-label without using any augmentation. When the confidence is low, we apply additionally N random augmentations to further improve the pseudo-label quality. The filtering is critical as we observe that random augmentations

Algorithm 1 The proposed continual test-time adaptation

Initialization: A source pre-trained model $f_{\theta_0}(x)$, teacher model $f_{\theta'_0}(x)$ initialized from $f_{\theta_0}(x)$.

Input: For each time step t , current stream of data x_t .

- 1: Augment x_t and get weight and augmentation-averaged pseudo-labels from the teacher $f_{\theta'_t}$ by Equation 4.
- 2: Update student f_{θ_t} by consistency loss in Equation 5.
- 3: Update teacher $f_{\theta'_t}$ by moving average in Equation 2.
- 4: Stochastically restore student f_{θ_t} by Equation 8.

Output: Prediction $f_{\theta'_t}(x_t)$; Updated student model $f_{\theta_{t+1}}(x)$; Updated teacher model $f_{\theta'_{t+1}}(x)$.

on confident samples with small domain gaps can sometimes decrease model performance. We provide detailed discussion on this observation in the supplementary. In summary, we use the confidence to approximate the domain difference and determine when to apply the augmentations.

The student is updated by the refined pseudo-label:

$$\mathcal{L}_{\theta_t}(x_t^T) = - \sum_c y_{tc}^T \log \hat{y}_{tc}^T, \quad (5)$$

Stochastic Restoration While more accurate pseudo-labels can mitigate error accumulation, continual adaptation by self-training for a long time inevitably introduces errors and leads to forgetting. This issue can be especially relevant if we encounter strong domain shifts within a sequence of data, because the strong distribution shift leads to mis-calibrated and even wrong predictions. Self-training in this case may only lead to reinforcing wrong predictions. What's worse is that after encountering hard examples, the model may not be able to recover because of the continual adaptation, even when the new data are not severely shifted.

To further tackle the problem of catastrophic forgetting, we propose a stochastic restoration method which explicitly restores the knowledge from the source pre-trained model.

Consider a convolution layer within the student model f_{θ} after gradient update based on Equation 1 at time step t :

$$x_{l+1} = W_{t+1} * x_l, \quad (6)$$

where $*$ denotes the convolution operation, x_l and x_{l+1} denote the input and output to this layer, W_{t+1} denotes the trainable convolution filters. The proposed stochastic restoration method additionally updates the weight W by:

$$M \sim \text{Bernoulli}(p), \quad (7)$$

$$W_{t+1} = M \odot W_0 + (1 - M) \odot W_{t+1}, \quad (8)$$

where \odot denotes the element-wise multiplication. p is a small restore probability, and M is a mask tensor of the same shape as W_{t+1} . The mask tensor decides which element within W_{t+1} to restore back to the source weight W_0 .

The stochastic restoration can also be seen as a special form of Dropout. By stochastically restoring a small number of tensor elements in the trainable weights to the initial weight, the network avoids drifting too far away from the initial source model and therefore, avoids catastrophic forgetting. In addition, by preserving the information from the source model, we are able to train all trainable parameters without suffering from model collapse. This brings more capacity for the adaptation and is another major difference compared to entropy minimization methods [43, 61] which only train the BN parameters for test-time adaptation.

As shown in Algorithm 1, combining the refined pseudo-labels with stochastic restoration leads to our online continual test-time adaptation (CoTTA) method.

4. Experiments

We evaluate our proposed method on five continual test-time adaptation benchmark tasks: CIFAR10-to-CIFAR10C (standard and gradual), CIFAR100-to-CIFAR100C, and ImageNet-to-ImageNet-C for image classification, as well as Cityscapes-to-ACDC for semantic segmentation.

4.1. Datasets and tasks

CIFAR10C, CIFAR100C, and ImageNet-C were originally created to benchmark robustness of classification networks [15]. Each dataset contains 15 types of corruptions with 5 levels of severity. The corruptions were applied on images from the test set of the clean CIFAR10 or CIFAR100 dataset [25]. There are 10,000 images for each corruption type for both CIFAR10C and CIFAR100C datasets.

For our online continual test-time adaptation task, a network pre-trained on the clean training set of CIFAR10 or CIFAR100 dataset is used. During test time, the corrupted images are provided in an online fashion to the network. Unlike previous methods which evaluate the test-time adaptation performance from the clean images pre-trained model to each corruption type individually, we continually adapt the source pre-trained model to each corruption type sequentially. We evaluate all models under the largest corruption severity level 5. The evaluation is based on the online prediction results immediately after the encounter of the data. Both the CIFAR10 and CIFAR100 experiments follow this online continual test-time adaptation scheme.

For CIFAR10-to-CIFAR10C, we follow the official public implementation from TENT [61] for the CIFAR10 experiments. The same pre-trained model is adopted, which is a WideResNet-28 [71] model from the RobustBench benchmark [8]. We update the model for one step at each iteration (i.e. one gradient step per test point). We use the same Adam optimizer with a learning rate of $1e-3$ as the official implementation. Following [5], we use the same random augmentation composition including color jitter, random affine, gaussian blur, random horizontal flip, and gaussian

noise. We use 32 augmentations for our experiments. We discuss the choice of the augmentation threshold p_{th} in our supplementary material. Unlike TENT models which only update the BN scale and shift weights, we update all trainable parameters in the experiments. We use a restoration probability of $p = 0.01$ for all our experiments.

For CIFAR100-to-CIFAR100C experiments, we adopt the pre-trained ResNeXt-29 [65] model from [16], which is used as one of the default architectures for CIFAR100 in the RobustBench benchmark [8]. The same hyperparameters are used as in the CIFAR10 experiments. The ImageNet-to-ImageNet-C [15] experiments use the *standard pre-trained resnet50* model in RobustBench [8]. ImageNet-C experiments are evaluated under ten diverse corruption orders.

Cityscapes-to-ACDC is a continual semantic segmentation task we designed to mimic continual distribution shifts in the real world. The source model is an off-the-shelf pre-trained segmentation model trained on the Cityscapes dataset [7]. The target domain contains images from various scenarios from the Adverse Conditions Dataset (ACDC) [50]. The ACDC dataset shares the same semantic classes with Cityscapes and is collected in four different adverse visual conditions: Fog, Night, Rain, and Snow. We evaluate our continual test-time adaptation following the same default order. We use 400 unlabeled images from each adverse condition for the adaptation. To mimic the scenario in real life where similar environments might be revisited, and to evaluate the forgetting effect of our methods, we repeat the same sequence group (of the four conditions) 10 times (i.e. in total 40: Fog→Night→Rain→Snow→Fog...). This also provides an evaluation of the adaptation performance in the long term.

For the implementation details, we adopt a transformer-based architecture, Segformer [64], for our Cityscapes-to-ACDC experiments. We use the publicly-available pre-trained Segformer-B5 trained on Cityscapes as our off-the-shelf source model. For the baseline comparison method, *TENT* optimizes the parameters in the normalization layers. For the proposed CoTTA model, all trainable layers are updated without the need to choose specific layers. Images from ACDC have a resolution of 1920×1080 . We use down-sampled resolutions of 960×540 as inputs to the network and the predictions are evaluated under the original resolution. Adam optimizer is used with the learning rate 8 times smaller than the default one for Segformer, because we use batch size 1 instead of 8 (default for source training) in our online continual test-time adaptation experiments. We use the multi-scaling input with flip as the augmentation method for the proposed method to generate augmentation-weighted pseudo-label (as in Equation 3). Following the default practice designed for Cityscapes in MMSeg [6], we use the scale factors of [0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0].

Table 2. Classification error rate (%) for the standard CIFAR10-to-CIFAR10C online continual test-time adaptation task. Results are evaluated on WideResNet-28 with the largest corruption severity level 5. * denotes the requirement on additional domain information.

Method	Weight- avg.	Aug- avg.	Stochastic Restore	$t \rightarrow$															Mean
				Gaussian	shot	impulse	defocus	glass	motion	zoom	snow	frost	fog	brightness	contrast	elastic.trans	pixelate	jpeg	
Source				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 Stats Adapt				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
Pseudo-label				26.7	22.1	32.0	13.8	32.2	15.3	12.7	17.3	17.3	16.5	10.1	13.4	22.4	18.9	25.9	19.8
TENT-online* [61]				24.8	23.5	33.0	12.0	31.8	13.7	10.8	15.9	16.2	13.7	7.9	12.1	22.0	17.3	24.2	18.6
TENT-continual [61]				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
CoTTA (Ours)	✓			27.2	22.8	30.8	12.1	30.1	13.9	11.9	17.2	16.0	14.3	9.4	13.1	19.9	15.4	19.9	18.3
CoTTA (Ours)	✓	✓		24.5	21.0	26.0	12.3	27.9	13.9	12.0	16.6	15.9	14.7	9.4	13.6	19.8	14.7	18.7	17.4
CoTTA (Ours)	✓	✓	✓	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 (0.1)

Table 3. Gradually changing CIFAR10-to-CIFAR10C results. The severity level changes gradually between the lowest and the highest. The corruption type changes when the severity is the lowest. Results are the mean over ten diverse corruption type sequences.

Avg. Error (%)	Source	BN Adapt	TENT-continual [61]	CoTTA
CIFAR10C	24.8	13.7	30.7	10.4 ± 0.2

4.2. Experiments on CIFAR10-to-CIFAR10C

We first evaluate the effectiveness of the proposed model on the CIFAR10-to-CIFAR10C task. We compare our method to the source-only baseline and four popular methods. As shown in Table 2, directly using the pre-trained model without adaptation yields a high average error rate of 43.5%, indicating that an adaptation is necessary. The *BN Stats Adapt* method keeps the network weights and uses the Batch Normalization statistics from the input data of the current iteration for the prediction [34, 51]. The approach is simple and fully online, and significantly improves the performance over the source-only baseline. Using hard pseudo-labels [31] to update the BN-trainable parameters can reduce the error rate to 19.8%. If the *TENT-online* [61] method has access to the additional domain information and resets itself to the initial pre-trained model whenever it encounters a new domain, the performance can be further improved to 18.6%. However, such information is usually unavailable in real applications. Without having access to this additional information, the *TENT-continual* method does not yield any improvement over the *BN Stats Adapt* method. It is worth mentioning that in earlier stages of the adaptation, *TENT-continual* outperforms the *BN Stats Adapt*. However, the model quickly deteriorates after observing three types of corruptions. This indicates that *TENT* based methods can be unstable under continual adaptation in the long term because of error accumulation. Our proposed method can continuously outperform all the above methods by using the weight-and-augmentation-averaged consistency. The error rate is significantly reduced to 16.2%. In addition, it does not suffer from performance deterioration in the long term because of our stochastic restore approach.

Ablation study: individual components The main contribution of our proposed method is to reduce the error ac-

cumulation by using averaged pseudo-labels and random restoration. To validate our motivation, we conduct an ablation study on each of the elements of the proposed approach. As listed in Table 2, by using the *weight-averaged pseudo-labels* from the teacher model, the error rate is reduced from 20.7% to 18.3%. This indicates that the weight-averaged predictions are indeed more accurate than the direct predictions. By using multiple augmentations to further refine the weight-averaged predictions, we are able to further improve the performance to 17.4%. However, the performance is still deteriorating over time (e.g. comparing to *TENT-online** for *contrast*), indicating that even though the pseudo-labels are more accurate, error can still accumulate because of the inevitable wrong predictions. Finally, by using *stochastic restoration* to explicitly preserve the source knowledge, the long-term predictions can be largely improved. This leads to an improved error rate of 16.2%. The number in bracket is the standard deviation over 5 seeds.

Gradually changing setup. In the above standard setup, corruption types change abruptly in the highest severity, we now report the results for the gradual setup. We design the sequence by gradually changing severity for the 15 corruption types: $\dots \xrightarrow[t-1 \text{ and before}]{\text{change type}} \underbrace{1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1}_{\text{corruption type } t, \text{ gradually changing severity}} \xrightarrow[t+1 \text{ and on}]{\text{change type}} 1 \rightarrow 2 \dots$, where the severity level is the lowest (1) when corruption type changes, therefore, the type change is gradual. The distribution shift within each type is also gradual. We create 10 randomly shuffled orders for the corruption types t and then evaluate the methods using the average error rate over the ten diverse sequences. Table 3 shows that the proposed method outperforms competing methods, leading to an error rate of 10.4%, compared to TENT’s 30.7%.

4.3. Experiments on CIFAR100-to-CIFAR100C

To further demonstrate the effectiveness of the proposed method, we evaluate it on the more difficult CIFAR100-to-CIFAR100C task. The experimental results are summarized in Table 4. We compare our method with the source-only baseline, *BN stats adapt*, *Pseudo-label*, as well as the *TENT-continual* method. We observe that performance of

Table 4. Classification error rate (%) for the standard CIFAR100-to-CIFAR100C online continual test-time adaptation task. All results are evaluated on the ResNeXt-29 architecture with the largest corruption severity level 5.

Time	$t \longrightarrow$																
Method	<i>Gaussian</i>	<i>shot</i>	<i>impulse</i>	<i>defocus</i>	<i>glass</i>	<i>motion</i>	<i>zoom</i>	<i>snow</i>	<i>frost</i>	<i>fog</i>	<i>brightness</i>	<i>contrast</i>	<i>elastic-trans</i>	<i>pixelate</i>	<i>jpeg</i>	Mean	
Source	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 Stats Adapt	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	
Pseudo-label	38.1	36.1	40.7	33.2	45.9	38.3	36.4	44.0	45.6	52.8	45.2	53.5	60.1	58.1	64.5	46.2	
TENT-continual [61]	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	
CoTTA (Proposed)	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	

Table 5. Semantic segmentation results (mIoU in %) on the Cityscapes-to-ACDC online continual test-time adaptation task. We evaluate the four test conditions continually for ten times to evaluate the long-term adaptation performance. To save space, we only show the continual adaptation results in the first, fourth, seventh, and last round. Full results can be found in the supplementary material. All results are evaluated based on the Segformer-B5 architecture.

Time	$t \longrightarrow$																
Round	1				4				7				10				All
Condition	Fog	Night	rain	snow	Fog	Night	rain	snow	Fog	Night	rain	snow	Fog	Night	rain	snow	Mean
Source	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	56.7
BN Stats Adapt	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	52.0
TENT-continual [61]	69.0	40.2	60.1	57.3	66.5	36.3	58.7	54.0	64.2	32.8	55.3	50.9	61.8	29.8	51.9	47.8	52.3
CoTTA (Proposed)	70.9	41.2	62.4	59.7	70.9	41.0	62.7	59.7	70.9	41.0	62.8	59.7	70.8	41.0	62.8	59.7	58.6

Table 6. Average error of standard ImageNet-to-ImageNet-C experiments over 10 diverse corruption sequences (severity level 5).

Avg. Error (%)	Source	BN Adapt	Test Aug [5]	TENT [58]	CoTTA
ImageNet-C	82.4	72.1	71.4	66.5	63.0 \pm 1.8 (0.1)

the *TENT-continual* model deteriorates rapidly over time on the later corruption types because of the error accumulation and forgetting. Our method yields an absolute improvement of 2.9% error rate over *BN stats adapt*, and achieves 32.5%. More importantly, the improvement becomes larger over time, this indicates that the proposed method is able to learn from the unlabeled test images from the past streams to further improve the performance on the current test data.

4.4. Experiments on ImageNet-to-ImageNet-C

To provide a more comprehensive evaluation on the proposed method, ImageNet-to-ImageNet-C experiments are conducted over ten diverse corruption type sequences in severity level of 5. As shown in Table 6, CoTTA is able to continually outperform *TENT* and other competing methods. The number after \pm is the standard deviation over 10 diverse corruption type sequences.

4.5. Experiments on Cityscapes-to-ACDC

We additionally evaluate our method on the more complex continual test-time semantic segmentation Cityscapes-to-ACDC task. The experimental results are summarized in Table 5. The results demonstrate that our method is also effective for semantic segmentation tasks and is robust to the different choices of architectures. Our proposed method yields an absolute improvement of 1.9% mIoU over the baseline, and achieves 58.6% mIoU. It is worth mention-

ing that *BN Stats Adapt* and *TENT* do not perform well in this task and the performance deteriorates significantly over time. This is partly because both were specifically designed for networks with Batch Normalization layers, while there is only one Batch Normalization layer in Segformer and the majority of normalization layers in transformer models are based on LayerNorm [2]. Our method, however, does not rely on specific layers and can still be effective for this more complex task on a very different architecture. The improved performance is also largely maintained after being continually adapted for a relatively long term.

5. Conclusion

In this work, we focused on the continual test-time adaptation in non-stationary environments where the target domain distribution can continually change over time. To tackle the error accumulation and catastrophic forgetting in this setup, we proposed a novel method CoTTA which comprises two parts. Firstly, we reduced the error accumulation by using weight-averaged and augmentation-averaged predictions which are often more accurate. Secondly, to preserve the knowledge from the source model, we stochastically restored a small part of the weights to the source pre-trained weights. The proposed method can be incorporated in off-the-shelf pre-trained models without requiring any access to source data. The effectiveness of CoTTA was validated on four classification and one segmentation tasks. **Acknowledgement** The contributions of Qin Wang and Olga Fink were funded by the Swiss National Science Foundation Grant PP00P2.176878. This work is also funded by Toyota Motor Europe via the project TRACE-Zurich.

References

- [1] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *ICLR*, 2020. 3
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 8
- [3] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. In *ICLR Workshops*, 2018. 3
- [4] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, pages 627–636, 2019. 2
- [5] Gilad Cohen and Raja Giryes. Katana: Simple post-training robustness using test time augmentations. *arXiv preprint arXiv:2109.08191*, 2021. 5, 6
- [6] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mms Segmentation>, 2020. 6
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6
- [8] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *NeuIPS Datasets and Benchmarks Track*, 2021. 6
- [9] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, pages 113–123, 2019. 5
- [10] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *T-PAMI*, 2021. 3
- [11] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. In *ECCV*. Springer, 2020. 2
- [12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015. 2
- [13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, pages 1321–1330. PMLR, 2017. 2
- [14] 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 *ICCV*, pages 8340–8349, 2021. 3
- [15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019. 6
- [16] 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. *ICLR*, 2020. 3, 6
- [17] Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *CVPR*, pages 867–874, 2014. 3
- [18] 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 *ICML*, pages 1989–1998. PMLR, 2018. 2
- [19] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. *arXiv preprint arXiv:2111.14887*, 2021. 2
- [20] Minhao Hu, Tao Song, Yujun Gu, Xiangde Luo, Jieneng Chen, Yinan Chen, Ya Zhang, and Shaoting Zhang. Fully test-time adaptation for image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 251–260. Springer, 2021. 3
- [21] Xuefeng Hu, Gokhan Uzunbas, Sirius Chen, Rui Wang, Ashish Shah, Ram Nevatia, and Ser-Nam Lim. Mixnorm: Test-time adaptation through online normalization estimation. *arXiv preprint arXiv:2110.11478*, 2021. 3
- [22] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeuIPS*, 2021. 3
- [23] Neerav Karani, Ertunc Erdil, Krishna Chaitanya, and Ender Konukoglu. Test-time adaptable neural networks for robust medical image segmentation. *Medical Image Analysis*, 68:101907, 2021. 3
- [24] 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
- [25] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Cite-seer, 2009. 6
- [26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeuIPS*, pages 1097–1105, 2012. 5
- [27] Jogendra Nath Kundu, Akshay Kulkarni, Amit Singh, Varun Jampani, and R Venkatesh Babu. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, pages 7046–7056, 2021. 3
- [28] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *CVPR*, pages 4544–4553, 2020. 2
- [29] Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *WACV*, pages 615–625, 2021. 2
- [30] Qicheng Lao, Xiang Jiang, Mohammad Havaei, and Yoshua Bengio. Continuous domain adaptation with variational domain-agnostic feature replay. *arXiv preprint arXiv:2003.04382*, 2020. 3

- [31] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 7
- [32] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. In *CVPR*, pages 12383–12392, 2021. 3
- [33] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pages 9641–9650, 2020. 2
- [34] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016. 3, 7
- [35] Zhizhong Li and Derek Hoiem. Learning without forgetting. *T-PAMI*, 40(12):2935–2947, 2017. 3
- [36] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*, October 2019. 2
- [37] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICLR*, pages 6028–6039. PMLR, 2020. 2
- [38] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, pages 1215–1224, 2021. 3
- [39] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015. 2
- [40] Alexander Lyzhov, Yuliya Molchanova, Arsenii Ashukha, Dmitry Molchanov, and Dmitry Vetrov. Greedy policy search: A simple baseline for learnable test-time augmentation. In Jonas Peters and David Sontag, editors, *Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 1308–1317. PMLR, 03–06 Aug 2020. 3
- [41] 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
- [42] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18. PMLR, 2013. 3
- [43] 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. 1, 2, 3, 6
- [44] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011. 2
- [45] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 2, 3
- [46] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 2
- [47] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992. 4, 5
- [48] Viraj Prabhu, Shivam Khare, Deeksha Kartik, and Judy Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *ICCV*, pages 8558–8567, 2021. 2
- [49] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 3
- [50] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, October 2021. 1, 6
- [51] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *NeurIPS*, 33, 2020. 7
- [52] Connor Shorten and Taghi M Khoshgohfar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019. 3, 5
- [53] Daniel L Silver and Robert E Mercer. The task rehearsal method of life-long learning: Overcoming impoverished data. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 90–101. Springer, 2002. 3
- [54] 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 *ICML*, pages 9229–9248. PMLR, 2020. 3, 4, 5
- [55] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017. 2, 4, 5
- [56] 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 *IROS*, pages 23–30. IEEE, 2017. 3
- [57] 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 *CVPR Workshops*, pages 969–977, 2018. 3
- [58] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 2
- [59] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. In *CVPR*, pages 4443–4453, 2021. 3

- [60] Dequan Wang, Shaoteng Liu, Sayna Ebrahimi, Evan Shelhamer, and Trevor Darrell. On-target adaptation. *arXiv preprint arXiv:2109.01087*, 2021. 3
- [61] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2, 3, 4, 6, 7, 8
- [62] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *ICCV*, pages 8515–8525, 2021. 2
- [63] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *ICRA*, pages 4489–4495. IEEE, 2018. 3
- [64] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 6
- [65] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 6
- [66] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, pages 8978–8987, 2021. 2
- [67] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, pages 4085–4095, 2020. 2
- [68] Hao-Wei Yeh, Baoyao Yang, Pong C Yuen, and Tatsuya Harada. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *WACV*, pages 474–483, 2021. 2
- [69] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *NeurIPS*, 2019. 3
- [70] Fuming You, Jingjing Li, and Zhou Zhao. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*, 2021. 3
- [71] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *BMVC*, 2016. 6
- [72] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, pages 3987–3995. PMLR, 2017. 3
- [73] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *arXiv preprint arXiv:2110.09506*, 2021. 3
- [74] Aurick Zhou and Sergey Levine. Training on test data with bayesian adaptation for covariate shift. *arXiv preprint arXiv:2109.12746*, 2021. 3
- [75] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, pages 289–305, 2018. 2