Robust Test-Time Adaptation in Dynamic Scenarios

Longhui Yuan Binhui Xie Shuang Li[✉]
School of Computer Science and Technology, Beijing Institute of Technology
{longhuiyuan,binhuixie,shuangli}@bit.edu.cn

Abstract

Test-time adaptation (TTA) intends to adapt the pretrained model to test distributions with only unlabeled test data streams. Most of the previous TTA methods have achieved great success on simple test data streams such as independently sampled data from single or multiple distributions. However, these attempts may fail in dynamic scenarios of real-world applications like autonomous driving, where the environments gradually change and the test data is sampled correlatively over time. In this work, we explore such practical test data streams to deploy the model on the fly, namely practical test-time adaptation (PTTA). To do so, we elaborate a **Ro***bust* **T***est-***T***ime* **A***daptation (RoTTA) method against the complex data stream in PTTA. More specifically, we present a robust batch normalization scheme to estimate the normalization statistics. Meanwhile, a memory bank is utilized to sample category-balanced data with consideration of timeliness and uncertainty. Further, to stabilize the training procedure, we develop a time-aware reweighting strategy with a teacher-student model. Extensive experiments prove that RoTTA enables continual test-time adaptation on the correlatively sampled data streams. Our method is easy to implement, making it a good choice for rapid deployment. The code is publicly available at* https://github.com/BIT-DA/RoTTA

1. Introduction

In recent years, many machine learning problems have made considerable headway with the success of deep neural networks [13, 22, 33, 38]. Unfortunately, the performance of deep models drops significantly when training data and testing data come from different distributions [59], which limits their utility in real-world applications. To reduce the distribution shift, a handful of works focus on transfer learning field [56], in particular, domain adaptation (DA) [17, 42, 45, 48, 69, 72] or domain generalization (DG) [40, 41, 52, 71, 83], in which one or more different but related labeled datasets (a.k.a. source domain) are collected to help the model generalize well to unlabeled or unseen samples in new datasets (a.k.a. target domain).

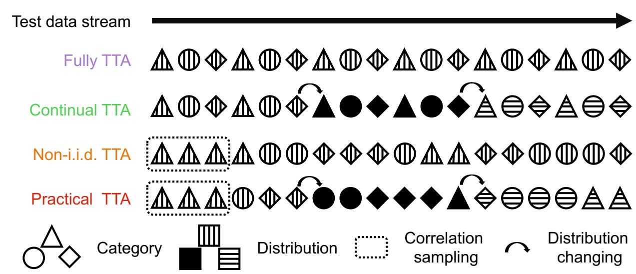


Figure 1. We consider the practical test-time adaptation (TTA) setup and compare it with related ones. First, Fully TTA [70] adapts models on a fixed test distribution with an independently sampled test stream. Then, on this basis, Continual TTA [73] takes the continually changing distributions into consideration. Next, Non-i.i.d. TTA [19] tries to tackle the correlatively sampled test streams on a single test distribution, where the label distribution among a batch of data deviates from that of the test distribution. To be more practical, Practical TTA strives to connect both worlds: distribution changing and correlation sampling.

While both DA and DG have extensively studied the problem of distribution shifts, they typically assume accessibility to the raw source data. However, in many practical scenarios like personal consumption records, the raw data should not be publicly available due to data protection regulations. Further, existing methods have to perform heavy backward computation, resulting in unbearable training costs. Test-time adaptation (TTA) [3,11,16,24,26,54,65,81] attempts to address the distribution shift online at test time with only unlabeled test data streams. Unequivocally, TTA has drawn widespread attention in a variety of applications, e.g., 2D/3D visual recognition [2, 29, 49, 65, 82], multimodality [63, 64] and document understanding [15].

Prior TTA studies [7, 20, 70, 73] mostly concentrate on a simple adaptation scenario, where test samples are independently sampled from a fixed target domain. To name a few, Sun *et al.* [65] adapt to online test samples drawn from a constant or smoothly changing distribution with an auxiliary self-supervised task. Wang *et al.* [70] adapt to a fixed

[✉]Corresponding author

Table 1. Comparison between our proposed practical test-time adaptation (PTTA) and related adaptation settings.

Setting	Adaptation Stage		Available Data		Test Data Stream	
	Train	Test	Source	Target	Distribution	Sampling Protocol
Domain Adaptation	✓	✗	✓	✓	-	-
Domain Generalization	✓	✗	✓	✗	-	-
Test-Time Training [65]	✓	✓	✓	✓	stationary	independently
Fully Test-Time Adaptation [70]	✗	✓	✗	✓	stationary	independently
Continual Test-Time Adaptation [73]	✗	✓	✗	✓	continually changing	independently
Non-i.i.d. Test-Time Adaptation [5, 19]	✗	✓	✗	✓	stationary	correlatively
Practical Test-Time Adaptation (Ours)	✗	✓	✗	✓	continually changing	correlatively

target distribution by performing entropy minimization online. However, such an assumption is violated when the test environments change frequently [73]. Later on, Boudiaf *et al*. [5] and Gong *et al*. [19] consider the temporal correlationship within test samples. For example, in autonomous driving, test samples are highly correlated over time as the car will follow more vehicles on the highway or will encounter more pedestrians in the streets. More realistically, the data distribution changes as the surrounding environment alerts in weather, location, or other factors. In a word, distribution change and data correlation occur simultaneously in reality.

Confronting continually changing distributions, traditional algorithms like pseudo labeling or entropy minimization become more unreliable as the error gradients cumulate. Moreover, the high correlation among test samples results in the erroneous estimation of statistics for batch normalization and collapse of the model. Driven by this analysis, adapting to such data streams will encounter two major obstacles: 1) incorrect estimation in the batch normalization statistics leads to erroneous predictions of test samples, consequently resulting in invalid adaptation; 2) the model will easily or quickly overfit to the distribution caused by the correlative sampling. Thus, such dynamic scenarios are pressing for a new TTA paradigm to realize robust adaptation.

In this work, we launch a more realistic TTA setting, where distribution changing and correlative sampling occur simultaneously at the test phase. We call this *Practical Test-Time Adaptation*, or briefly, *PTTA*. To understand more clearly the similarities and differences between PTTA and the previous setups, we visualize them in Figure 1 and summarize them in Table 1. To conquer this challenging problem, we propose a **Ro**bust **T**est-**T**ime **A**daptation (**RoTTA**) method, which consists of three parts: 1) robust statistics estimation, 2) category-balanced sampling considering timeliness and uncertainty and 3) time-aware robust training. More concretely, we first replace the erroneous statistics of the current batch with global ones maintained by the exponential moving average. It is a more stable manner to estimate the statistics in BatchNorm layers. Then, we simulate a batch of independent-like data in memory with category-balanced sampling while considering the timeliness and uncertainty of the buffered samples. That is, samples that are newer and less uncertain are kept in memory with higher priority. With this batch of category-balanced, timely and confident samples, we can obtain a snapshot of the current distribution. Finally, we introduce a time-aware reweighting strategy that considers the timeliness of the samples in the memory bank, with a teacher-student model to perform robust adaptation. With extensive experiments, we demonstrate that RoTTA can robustly adapt in the practical setup, i.e., PTTA.

In a nutshell, our contributions can be summarized as:

- We propose a new test-time adaptation setup that is more suitable for real-world applications, namely practical test-time adaptation (PTTA). PTTA considers both distribution changing and correlation sampling.
- We benchmark the performance of prior methods in PTTA and uncover that they only consider one aspect of the problem, resulting in ineffective adaptation.
- We propose a robust test-time adaptation method (RoTTA), which has a more comprehensive consideration of PTTA challenges. Ease of implementation and effectiveness make it a practical deployment option.
- We extensively demonstrate the practicality of PTTA and the effectiveness of RoTTA on common TTA benchmarks [23], i.e., CIFAR-10-C and CIFAR-100-C and a large-scale DomainNet [58] dataset. RoTTA obtains state-of-the-art results, outperforming the best baseline by a large margin (reducing the averaged classification error by over 5.9%, 5.5% and 2.2% on CIFAR-10-C, CIFAR-100-C and DomainNet, respectively).

2. Related Work

Domain adaptation (DA) studies the problem of transferring the knowledge learned from a labeled source dataset to an unlabeled target dataset [8, 17, 43, 51, 67, 68]. Representative techniques include latent distribution alignment [48, 77], adversarial training [17, 62], or self-training [75, 85]. The limitation of this setting, however, is that an unlabeled test dataset (target domain) is needed at training time, in addition to a labeled training dataset (source domain). Accordingly, it might fail to handle more practical scenarios

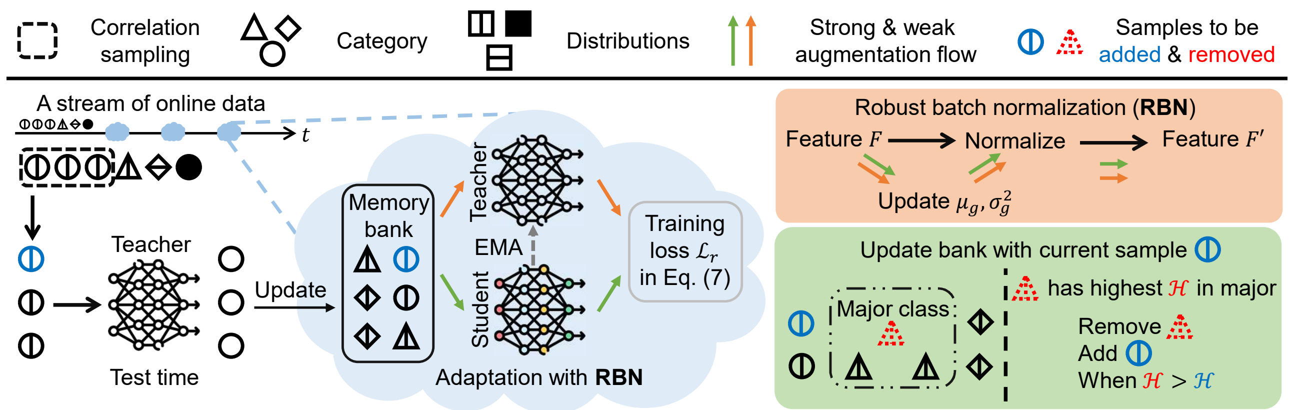


Figure 2. **Framework overview.** Firstly, we replace the batch normalization layer with RBN which robustly normalizes the feature map. During the inference of the online test stream of PTTA, we utilize the predictions of samples to maintain a memory bank by category-balanced sampling with timeliness and uncertainty. Finally, we use the category-balanced, timely and confident data in the memory bank combined with a robust loss to adapt the model at test time.

like test-time adaptation. Our practical test-time adaptation setting can be viewed as performing correlatively sample adaptation on the fly. It is worth noting that standard domain adaptation techniques might collapse when only continual data streams from multiple target domains are accessible.

Domain generalization (DG) assumes that multiple source domains are available for model training and tries to learn models that can generalize well to any unseen domains [4, 26, 40, 41, 52, 84]. A broad spectrum of methodologies based on data augmentation [78, 84], meta-learning [14, 40], or domain alignment [50, 52] has made great progress. In contrast, this work instead aims to improve the performance of source pre-trained models at the test time by using unlabeled online data streams from multiple continually changing target domains.

Continual learning (CL) (also known as incremental learning, life-long learning) addresses the problem of learning a model for many tasks sequentially without forgetting knowledge obtained from the preceding tasks. [1, 6, 31, 37, 60]. CL methods can often be categorized into replay-based [60, 66] and regularization-based [31, 44] methods. Ideas from continual learning are also adopted for continuous domain adaptation approaches [34, 74] In our work, we share the same motivation as CL and point out that practical test-time adaptation (PTTA) also suffers catastrophic forgetting (i.e., performance degradation on new test samples due to correlation sampling), which makes test-time adaptation approaches are unstable to deploy.

Test-time adaptation (TTA) focus on more challenging settings where only source model and unlabeled target data are available [9, 18, 27, 28, 35, 46, 61]. A similar paradigm is source-free domain adaptation (SFDA) [10, 36, 47, 79], which also requires no access to the training (source) data. To name a few, Liang *et al.* [45] fit the source hypothesis by exploiting the information maximization and self-supervised pseudo-labeling. Kundu *et al.* [35] formalize a unified solution that explores SFDA without any category-gap knowledge. To fully utilize any arbitrary pre-trained model, Sun *et al.* [65] propose conducting adaptation on the fly with an auxiliary self-supervised task. Later on, Wang *et al.* [70] take a source pre-trained model and adapt it to the test data by updating a few trainable parameters in BatchNorm layers [25] using entropy minimization [21].

While standard TTA has been widely studied in many tasks [2, 20, 63, 64, 70, 82], the fact remains that both distribution changing [73] and data correlation sampling [19] has only been considered in isolation. For example, Gong *et al.* [19] propose instance-aware batch normalization and prediction-balanced reservoir sampling to address the challenges of correlatively sampled test streams, however, it does not consider unstable adaptation resulting from long-term adaptation on continually changing distributions. On the other hand, Wang *et al.* [73] assume that the target test data is streamed from a continually changing environment and continually adapt an off-the-shelf source pre-trained model to the current test data. In this work, we launch PTTA, a more practical TTA setting to connect both worlds: distribution changing and correlation sampling.

3. Method

3.1. Problem Definition and Motivation

Given a model f_{θ_0} with parameter θ_0 pre-trained on source domain $\mathcal{D}_\mathcal{S} = \{(x^\mathcal{S}, y^\mathcal{S})\}$, the proposed practical test-time adaptation (PTTA) aims to adapt f_{θ_0} to a stream of online unlabeled samples $\mathcal{X}_0, \mathcal{X}_1, ..., \mathcal{X}_T$, where $\mathcal{X}_t$ is a batch of highly correlated samples from the distribution $\mathcal{P}_{test}$ that changes with time t continually. More specifically, at test time, with time going on, the test distribution $\mathcal{P}_{test}$ changes continually as $\mathcal{P}_0, \mathcal{P}_1, ..., \mathcal{P}_\infty$. At time step t, we will receive a batch of unlabeled and correlated samples

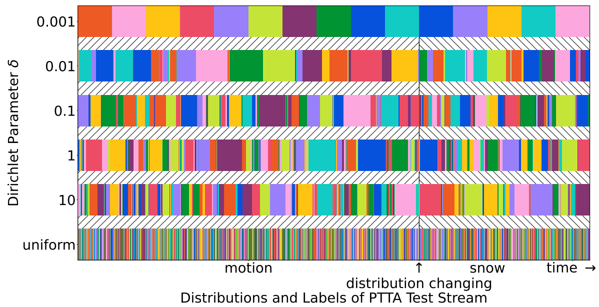


Figure 3. Illustration of the labels and distributions of the test stream of CIFAR10-C under the setup PTTA. And we adopt Dirichlet distribution to simulate the process of correlative sampling. It is clear that as the concentration parameter δ decreases, the correlation among sampled data increases, which is reflected in the increasing aggregation of categories.

$\mathcal{X}_t$ from $\mathcal{P}_{test}$. Next, $\mathcal{X}_t$ is fed into the model f_{θ_t} and the model needs to adapt itself to the current test data streams and make predictions $f_{\theta_t}(\mathcal{X}_t)$ on the fly.

As a matter of fact, this setup is largely driven the practical demands of deploying models in dynamic scenarios. Taking for example the case of autonomous driving mentioned in § 1, test samples are highly correlated and the data distribution changes continually with the weather or location. Another example is the situation of intelligent monitoring, the camera will continuously capture more people at certain times, such as after work, but fewer of them during work time. Meanwhile, the light condition changes continually from day to night. The deployed model should be robustly adapted in such dynamic scenarios. In a word, distribution change and data correlation often happen simultaneously in the real world. For this reason, existing TTA methods [7,9,19,28,70,73,81] might become unstable when the test stream is sampled from such dynamic scenarios.

To obtain the test stream of PTTA, we adopt Dirichlet Distribution with parameter δ to simulate the correlation among test samples. We present the test data streams corresponding to different values of δ on the CIFAR10-C dataset in Figure 3. We can observe that the smaller δ is, the higher the correlation will be. For the sake of unity, we set $\delta = 0.1$ as the default for all experiments. In the following, we present a robust test-time adaptation framework for the practical test-time adaptation setup defined above. An overview of our RoTTA is illustrated in Figure 2.

3.2. Robust Test-Time Adaptation

Motivated by the fact that the statistics of current batch data, which are commonly used in previous TTA methods [7, 20, 65, 70, 73], become unreliable when they encounter correlative test data streams, we first turn to the global robust statistics for normalization. Then, to effectively adapt to the current distribution, we maintain a memory bank by category-balanced sampling with considering timeliness and uncertainty, which captures a more stable snapshot of the distribution. Finally, we utilize the teacher-student model and design a timeliness-based reweighting strategy to train the model robustly.

Robust batch normalization (RBN). Batch Normalization (BN) [25] is a widely-used training technique as it can accelerate the training and convergence speed of networks and stabilize the training process by reducing the risk of gradient explosion and vanishing. Given the feature map $F \in \mathbb{R}^{B\times C\times H\times W}$ as the input for a BN layer when training, the channel-wise mean $\mu \in \mathbb{R}^C$ and variance $\sigma^2 \in \mathbb{R}^C$ are calculated as follows:

$$\mu_c = \frac{1}{BHW}\sum_{b=1}^{B}\sum_{h=1}^{H}\sum_{w=1}^{W} F_{(b,c,h,w)}\,, \tag{1}$$

$$\sigma_c^2 = \frac{1}{BHW}\sum_{b=1}^{B}\sum_{h=1}^{H}\sum_{w=1}^{W} (F_{(b,c,h,w)} - \mu_c)^2\,. \tag{2}$$

Then the feature map is normalized and refined in a channel-wise manner as

$$BN(F_{(b,c,h,w)};\mu,\sigma^2) = \gamma_c \frac{F_{(b,c,h,w)} - \mu_c}{\sqrt{\sigma_c^2+\epsilon}} + \beta_c\,, \tag{3}$$

where $\gamma, \beta \in \mathbb{R}^C$ are learnable parameters in the layer and $\epsilon > 0$ is a constant for numerical stability. Meanwhile, during training, the BN layer maintains a group of global running mean and running variance (μ_s, σ_s^2) for inference.

Due to the domain shift at test time, the global statistics (μ_s, σ_s^2) normalize test features inaccurately, causing significant performance degradation. To tackle the problem above, some methods [55, 70, 73] use the statistics of the current batch to perform normalization. Unfortunately, when the test samples have a high correlation under PTTA setup, the statistics of the current batch also fail to correctly normalize the feature map, as demonstrated in Figure 4c. Specifically, the performance of BN [53] decreases rapidly as the data correlation increases.

Based on the analysis above, we propose a robust batch normalization (RBN) module, which maintains a group of global statistics (μ_g, σ_g^2) to normalize the feature map robustly. Before the whole test-time adaptation, (μ_g, σ_g^2) is initialized as the running mean and variance (μ_s, σ_s^2) of the pre-trained model. When adapting the model, we update the global statistics first by exponential moving average as

$$\mu_g = (1-\alpha)\mu_g + \alpha\mu\,, \tag{4}$$

$$\sigma_g^2 = (1-\alpha)\sigma_g^2 + \alpha\sigma^2\,, \tag{5}$$

where (μ, σ^2) is the statistics of the buffered samples in the memory bank. Then we normalize and affine the feature as Eq. (3) with (μ_g, σ_g^2). When inferring for test samples, we directly utilize (μ_g, σ_g^2) to calculate the output as Eq (3). Although simple, RBN is effective enough to tackle the problem of normalization on test streams of PTTA.

Category-balanced sampling with timeliness and uncertainty (CSTU). In the PTTA setup, the correlation among test samples $\mathcal{X}_t$ at time t leads to a deviation between the observed distribution $\widehat{\mathcal{P}}_{test}$ and the test distribution $\mathcal{P}_{test}$. Specifically, the marginal label distribution $p(y|t)$ tends to differ from $p(y)$. Continuously learning with $\mathcal{X}_t$ over time t can lead to model adaptation to an unreliable distribution $\widehat{\mathcal{P}}_{test}$, resulting in ineffective adaptation and an increased risk of model collapse.

To address this issue, we propose a category-balanced memory bank $\mathcal{M}$ with a capacity of $\mathcal{N}$, which takes into account the timeliness and uncertainty of samples when updating. In particular, we adopt the predictions of test samples as pseudo labels to guide the update of $\mathcal{M}$. Meanwhile, to guarantee the balance among categories, we distribute the capacity of $\mathcal{M}$ equally to each category, and samples of the major categories will be replaced first (refer to lines 5-9 in Algorithm 1). Furthermore, due to the continually changing test distribution, old samples in $\mathcal{M}$ are limited in value, and could even impair the ability of the model to adapt to the current distribution. Additionally, samples of high uncertainty always produce erroneous gradient information that can hinder model adaptation, as suggested by [55].

With this in mind, we attach each sample in $\mathcal{M}$ with a group of heuristics $(\mathcal{A}, \mathcal{U})$, where $\mathcal{A}$, initialized as 0 and increasing with time t, is the age of the sample, and $\mathcal{U}$ the uncertainty calculated as the entropy of the prediction. Next, we combine the timeliness and uncertainty to calculate a heuristic score, i.e., category-balanced sampling with timeliness and uncertainty (CSTU), as follows:

$$\mathcal{H} = \lambda_t \frac{1}{1 + \exp(-\mathcal{A}/\mathcal{N})} + \lambda_u \frac{\mathcal{U}}{\log \mathcal{C}} , \tag{6}$$

where λ_t and λ_u make the trade-off between timeliness and uncertainty, and for simplicity, λ_t and λ_u are set to 1.0 for all experiments, and $\mathcal{C}$ is the number of categories. We summarize our sampling algorithm in Algorithm 1. With CSTU, we can obtain a robust snapshot of the current test distribution $\mathcal{P}_{test}$, and effectively adapt the model to it.

Robust training with timeliness. Actually, after replacing BN layers with our RBN and obtaining the memory bank selected via CSTU, we can directly adopt the widely used techniques like pseudo labeling or entropy minimization to perform test-time adaptation. However, we notice that too old or unreliable instances still have the opportunity to stay in $\mathcal{M}$ since keeping the category balance is assigned the top priority. In addition, too aggressive updates of the model will make the category balance of $\mathcal{M}$ unreliable, resulting in unstable adaptation. Meanwhile, error accumulation caused by the distribution change also makes the aforementioned approaches unworkable.

To further reduce the risk of error gradients information from old and unreliable instances and stabilize the adaptation, we turn to the robust unsupervised learning method

---

Algorithm 1: CSTU for one test sample.

```
1  Input: a test sample x and the teacher model f_{θ^T}.
2  Define: memory bank M and its capacity N,
     number of classes C, per class occupation O ∈ R^C,
     total occupation Ω, classes to pop instance D.
3  Infer as p(y|x) = Softmax(f_{θ^T}(x)).
4  Calculate the predicted category of x as
     ŷ = arg max_c p(c|x), the uncertainty as
     U_x = −Σ_{c=1}^{C} p(c|x) log(p(c|x)), the age as
     A_x = 0, and the heuristic score H_x of x with
     Eq (6)
5  if O_ŷ < N/C then
6      if Ω < N: Search range D = ∅.
7      else: Search range D = {j | j = arg max_c O_c}
8  else
9      Search range D = {ŷ}
10 if D is ∅ then
11     Add (x, ŷ, H_x, U_x) into M.
12 else
13     Find the instance (x̂, y_x̂, A_x̂, U_x̂) with the
         highest value in Eq (6) H_x̂ among D.
14     if H_x < H_x̂ then
15         Remove (x̂, y_x̂, A_x̂, U_x̂) from M.
16         Add (x, ŷ, H_x, U_x) into M.
17     else
18         Discard x.
19 Increase the age of all instances in M.
```

---

teacher-student model and propose a timeliness reweighting strategy. In addition, for the sake of time efficiency and stability, only affine parameters in RBN are trained during adaptation.

At time step t, after inferring for the correlated data $\mathcal{X}_t$ with the teacher model $f_{\theta_t^T}$ and updating the memory bank $\mathcal{M}$ with $\mathcal{X}_t$, we begin updating the student model $f_{\theta_t^S}$ and the teacher model $f_{\theta_t^T}$. Firstly, we update parameters of student model $\theta_t^S \rightarrow \theta_{t+1}^S$ by minimizing the following loss:

$$\mathcal{L}_r = \frac{1}{\Omega} \sum_{i=1}^{\Omega} \mathcal{L}(x_i^{\mathcal{M}}, \mathcal{A}_i; \theta_t^T, \theta_t^S) , \tag{7}$$

where $\Omega = |\mathcal{M}|$ is the total occupation of the memory bank, and $x_i^{\mathcal{M}}$ and $\mathcal{A}_i (i = 1, ..., \Omega)$ are instances in the memory bank and their age respectively. Subsequently, the teacher model is updated by exponential moving average as

$$\theta_{t+1}^T = (1 - \nu)\theta_t^T + \nu\theta_{t+1}^S . \tag{8}$$

To calculate the loss value of an instance $x_i^{\mathcal{M}}$ from the memory bank, the timeliness reweighting term is computed as

$$E(\mathcal{A}_i) = \frac{\exp(-\mathcal{A}_i/\mathcal{N})}{1 + \exp(-\mathcal{A}_i/\mathcal{N})} , \tag{9}$$

where $\mathcal{A}_i$ is the age of $x_i^{\mathcal{M}}$, and $\mathcal{N}$ is the capacity of the bank. And then we calculate the cross entropy between the soft-max prediction $p_S(y|x_i'')$ of the strong-augmented view x_i'' from the student model and that $p_T(y|x_i')$ of the weak-augmented view [1] x_i' from the teacher model as follows:

$$\ell(x_i', x_i'') = -\frac{1}{\mathcal{C}} \sum_{c=1}^{\mathcal{C}} p_T(c|x_i') \log p_S(c|x_i'') \,. \tag{10}$$

Finally, equipped with Eq. (9) and Eq. (10), the right-hand side of Eq. (7) reduces to

$$\mathcal{L}(x_i^{\mathcal{M}}, \mathcal{A}_i; \theta_t^T, \theta_t^S) = E(\mathcal{A}_i)\ell(x_i', x_i'') \,. \tag{11}$$

To sum up, equipped with RBN, CSTU, and robust training with timeliness, our RoTTA is capable of effectively adapting any pre-trained models in dynamic scenarios.

4. Experiments

4.1. Setup

Datasets. CIFAR10-C and CIFAR100-C [23] are the commonly used TTA benchmarks to testify the robustness under corruptions. Both of them are obtained by applying 15 kinds of corruption with 5 different degrees of severity on their clean test images of original datasets CIFAR10 and CIFAR100 respectively. CIFAR10/CIFAR100 [32] have 50,000/10,000 training/test images, all of which fall into 10/100 categories. DomainNet [58] is the largest and hardest dataset to date for domain adaptation and consists of about 0.6 million images with 345 classes. It consists of six different domains including Clipart (clp), Infograph (inf), Painting (pnt), Quickdraw (qdr), Real (rel), and Sketch (skt). We first pre-train a source model on the train set in one of six domains and testify all baseline methods on the test set of the remaining five domains.

Implementation details. All experiments are conducted with PyTorch [57] framework. In the case of robustness to corruption, following the previous methods [55, 70, 73], we obtain the pre-trained model from RobustBench benchmark [12], including the WildResNet-28 [80] for CIFAR10 $\rightarrow$ CIFAR10-C, and the ResNeXt-29 [76] for CIFAR100 $\rightarrow$ CIFAR100-C. Then, we change the test corruption at the highest severity 5 one by one to simulate that the test distribution continually changes with time in PTTA. And in the case of generalization under the huge domain gap, we train a ResNet-101 [22] by standard classification loss for each domain in DomainNet and adapt them continually to different domains except the source domain. Meanwhile, we utilize the Dirichlet distribution to simulate the correlatively sampled test stream for all datasets. For optimization, we adopt Adam [30] optimizer with learning rate 1.0×10^{-3}, $\beta = 0.9$. For a fair comparison, we set the batch size for all methods as 64 and the capacity of the memory bank of RoTTA as $\mathcal{N} = 64$. Concerning the hyperparameters, we adopt a unified set of values for RoTTA across all experiments including $\alpha = 0.05$, $\nu = 0.001$, $\lambda_t = 1.0$, $\lambda_u = 1.0$, and $\delta = 0.1$. More details are provided in the appendix.

4.2. Comparisons with the State-of-the-arts

Robustness under corruptions. The classification error on CIFAR10→CIFAR10-C and CIFAR100→CIFAR100-C are shown in Table 2 and Table 3 respectively. We change the type of the current corruption at the highest severity 5 as time goes on, and sample data correlatively for inference and adaptation simultaneously. The same test stream is shared across all compared methods.

From Table 2 and Table 3, we can see that RoTTA achieves the best performance compared to previous methods. Moreover, RoTTA has a significant performance gain to the second-best method that 5.9% improvement on CIFAR10→CIFAR10-C and 5.5% improvement on CIFAR100→CIFAR100-C respectively, verifying the effectiveness of RoTTA to adapt the model under PTTA.

In more detail, we can observe that BN [53], PL [39], TENT [70] and CoTTA [73] negatively adapt the model to the test streams of both datasets compared to Source ($-6.5 \sim -46.4\%$). This is attributed to the fact that these methods overlook the issues posed by correlation sampling, which can result in highly correlated data within a batch. As a consequence, traditional normalization statistics may be ineffective in appropriately normalizing the feature maps. Equipped with RBN and CSTU, RoTTA no longer suffers from this issue. Meanwhile, in Table 3, if focus on the adaptation procedure, we can see that the performance of PL [39], TENT [70] and NOTE [19] becomes worse and worse, and eventually, the model even collapses (error rate $> 97\%$). This reveals that the impact of error accumulation on long-term adaptation can be catastrophic. To tackle this problem, RoTTA turns to robustly adapt the model with timeliness reweighting and confident samples in the memory bank, and superior performance throughout the adaptation process demonstrates its effectiveness.

In addition, we find that although LAME [5] never tunes the parameters of the model, it is still a competitive baseline for example it achieves the second-best result on CIFAR100→CIFAR100-C. However, its performance is very dependent on the performance of the pre-trained model e.g. negligible improvement on difficult corruptions (shot, gaussian, pixelate). On the contrary, our RoTTA is more flexible and achieves better and more robust results.

Generalization under domain shift. We also evaluate RoTTA under a more challenging dataset DomainNet, where we continually adapt a source pre-trained model to correlatively sampled test streams of the rest domains. As

[1] Weak augmentation is ReSize+CenterCrop. Strong augmentation is a combination nine operations like Clip, ColorJitter, and RandomAffine.

Table 2. Average classification error of the task CIFAR10 → CIFAR10-C while continually adapting to different corruptions at the highest severity 5 with correlatively sampled test stream under the proposed setup PTTA.

Time	t →															
Method	motion	snow	fog	shot	defocus	contrast	zoom	brightness	frost	elastic	glass	gaussian	pixelate	jpeg	impulse	Avg.
Source	34.8	25.1	26.0	65.7	46.9	46.7	42.0	9.3	41.3	26.6	54.3	72.3	58.5	30.3	72.9	43.5
BN [53]	73.2	73.4	72.7	77.2	73.7	72.5	72.9	71.0	74.1	77.7	80.0	76.9	75.5	78.3	79.0	75.2
PL [39]	73.9	75.0	75.6	81.0	79.9	80.6	82.0	83.2	85.3	87.3	88.3	87.5	87.5	87.5	88.2	82.9
TENT [70]	74.3	77.4	80.1	86.2	86.7	87.3	87.9	87.4	88.2	89.0	89.2	89.0	88.3	89.7	89.2	86.0
LAME [5]	29.5	**19.0**	20.3	65.3	42.4	43.4	36.8	**5.4**	37.2	**18.6**	51.2	73.2	57.0	**22.6**	71.3	39.5
CoTTA [73]	77.1	80.6	83.1	84.4	83.9	84.2	83.1	82.6	84.4	84.2	84.5	84.6	82.7	83.8	84.9	83.2
NOTE [19]	**18.0**	22.1	20.6	35.6	26.9	**13.6**	26.5	17.3	27.2	37.0	48.3	38.8	42.6	41.9	49.7	31.1
RoTTA	18.1	21.3	**18.8**	**33.6**	**23.6**	16.5	**15.1**	11.2	**21.9**	30.7	**39.6**	**26.8**	**33.7**	27.8	**39.5**	**25.2**(+5.9)

Table 3. Average classification error of the task CIFAR100 → CIFAR100-C while continually adapting to different corruptions at the highest severity 5 with correlatively sampled test stream under the proposed setup PTTA.

Time	t →															
Method	motion	snow	fog	shot	defocus	contrast	zoom	brightness	frost	elastic	glass	gaussian	pixelate	jpeg	impulse	Avg.
Source	30.8	39.5	50.3	68.0	29.3	55.1	28.8	29.5	45.8	37.2	54.1	73.0	74.7	41.2	39.4	46.4
BN [53]	48.5	54.0	58.9	56.2	46.4	48.0	47.0	45.4	52.9	53.4	57.1	58.2	51.7	57.1	58.8	52.9
PL [39]	50.6	62.1	73.9	87.8	90.8	96.0	94.8	96.4	97.4	97.2	97.4	97.4	97.3	97.4	97.4	88.9
TENT [70]	53.3	77.6	93.0	96.5	96.7	97.5	97.1	97.5	97.3	97.2	97.1	97.7	97.6	98.0	98.3	92.8
LAME [5]	**22.4**	**30.4**	43.9	66.3	**21.3**	51.7	**20.6**	**21.8**	39.6	**28.0**	48.7	72.8	74.6	**33.1**	**32.3**	40.5
CoTTA [73]	49.2	52.7	56.8	53.0	48.7	51.7	49.4	48.7	52.5	52.2	54.3	54.9	49.6	53.4	56.2	52.2
NOTE [19]	45.7	53.0	58.2	65.6	54.2	52.0	59.8	63.5	74.8	91.8	98.1	98.3	96.8	97.0	98.2	73.8
RoTTA	31.8	36.7	**40.9**	**42.1**	30.0	**33.6**	27.9	25.4	**32.3**	34.0	**38.8**	**38.7**	**31.3**	38.0	42.9	**35.0**(+5.5)

Table 4. Average classification error of DomainNet while continually adapting to different domains with correlatively sampled test stream.

Time	t →						
Source	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	83.9	65.4	88.6	48.0	59.1	69.0
inf	61.8	N/A	66.9	96.0	50.0	70.6	69.1
pnt	56.5	83.7	N/A	94.2	42.6	63.4	68.1
qdr	89.2	99.0	98.6	N/A	95.0	92.3	94.8
rel	49.4	80.4	51.5	93.4	N/A	63.3	67.6
skt	47.5	88.2	62.9	87.1	51.8	N/A	67.5
Avg.	60.9	87.0	69.1	91.9	57.5	69.7	72.7

Time	t →						
BN	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	88.6	70.7	90.5	65.4	67.0	76.5
inf	68.6	N/A	74.2	96.2	69.9	76.8	77.1
pnt	60.8	87.9	N/A	94.3	62.3	68.7	74.8
qdr	80.3	97.7	92.6	N/A	88.7	88.1	89.5
rel	57.9	87.1	63.1	94.3	N/A	70.8	74.6
skt	50.4	87.6	64.6	89.6	63.1	N/A	71.1
Avg.	63.6	89.8	73.0	93.0	69.9	74.3	77.3

Time	t →						
PL	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	94.5	98.9	99.5	99.7	99.7	98.5
inf	82.6	N/A	99.2	99.6	99.7	99.3	96.1
pnt	78.6	99.4	N/A	99.7	99.6	99.7	95.4
qdr	81.7	99.5	99.6	N/A	99.7	99.8	96.1
rel	73.5	99.4	99.2	99.6	N/A	99.7	94.3
skt	64.8	99.2	99.4	99.7	99.7	N/A	92.6
Avg.	76.2	98.4	99.3	99.6	99.7	99.6	95.5

Time	t →						
TENT	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	87.5	71.9	94.2	96.2	98.9	89.7
inf	68.6	N/A	75.0	97.3	95.9	98.7	87.1
pnt	61.7	87.1	N/A	96.4	95.3	98.8	87.8
qdr	78.9	97.1	91.6	N/A	89.2	88.7	89.1
rel	57.8	86.4	68.1	96.9	N/A	96.7	81.2
skt	51.9	87.2	69.1	95.3	97.3	N/A	80.1
Avg.	63.8	89.0	75.1	96.0	94.8	96.4	85.8

Time	t →						
LAME	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	82.2	64.5	87.7	46.9	58.9	68.0
inf	60.1	N/A	65.7	95.4	48.5	69.4	67.8
pnt	55.8	81.5	N/A	93.3	41.3	62.1	66.8
qdr	88.3	99.1	99.0	N/A	94.9	92.2	94.7
rel	48.0	79.3	50.1	91.6	N/A	60.2	65.8
skt	45.6	87.1	59.5	83.9	49.9	N/A	65.2
Avg.	59.6	85.8	67.8	90.4	56.3	68.6	71.4

Time	t →						
COTTA	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	90.6	77.9	89.3	76.3	72.7	81.4
inf	74.5	N/A	82.0	95.7	80.2	81.5	82.8
pnt	66.3	89.8	N/A	93.4	74.0	75.4	79.8
qdr	82.3	98.2	94.6	N/A	92.5	90.1	91.5
rel	64.0	90.3	73.2	93.5	N/A	77.6	79.7
skt	56.1	89.2	71.9	89.2	73.5	N/A	76.0
Avg.	68.6	91.6	79.9	92.2	79.3	79.5	81.9

Time	t →						
NOTE	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	89.2	73.0	94.8	98.4	99.4	91.0
inf	75.4	N/A	78.7	98.7	98.1	99.5	90.1
pnt	64.7	89.8	N/A	97.8	98.4	99.2	90.0
qdr	74.7	97.2	92.2	N/A	93.5	99.6	91.4
rel	61.3	89.2	68.9	98.8	N/A	99.2	83.5
skt	55.2	89.7	70.1	96.9	98.3	N/A	82.0
Avg.	66.3	91.0	76.6	97.4	97.3	99.4	88.0

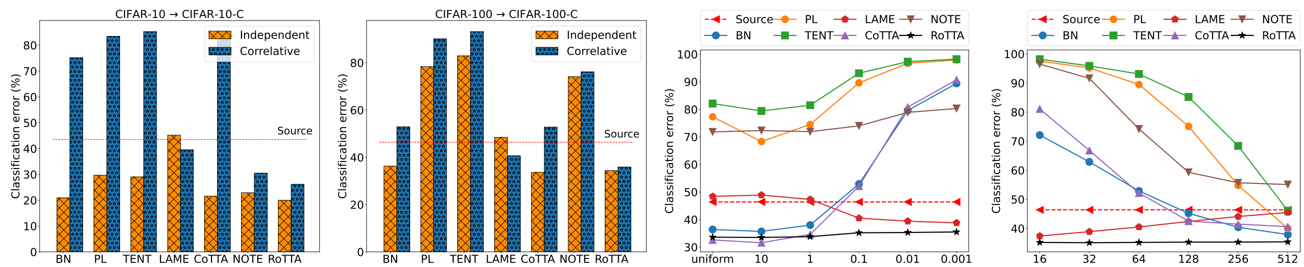
Time	t →						
RoTTA	*clp*	*inf*	*pnt*	*qdr*	*rel*	*skt*	Avg.
clp	N/A	85.5	62.0	82.0	49.3	59.8	67.7
inf	61.8	N/A	63.7	91.5	52.5	67.6	67.4
pnt	53.3	84.1	N/A	89.1	47.3	61.4	67.0
qdr	77.5	97.0	89.8	N/A	80.3	82.2	85.3
rel	49.1	82.3	50.3	88.0	N/A	61.1	66.2
skt	42.6	83.7	54.4	80.9	47.5	N/A	61.8
Avg.	56.8	86.5	64.0	86.3	55.4	66.4	**69.2**(+2.2)

shown in Table 4, consistent with the previous analysis, most of the methods include BN [53], PL [39], TENT [70], CoTTA [73] and NOTE [19] even perform worse than the Source model ($-4.6 \sim -22.8\%$). RoTTA consistently achieves the best performance and has 2.2% gain than the second method LAME [5], demonstrating RoTTA's effectiveness again.

4.3. Ablation Study

Effect of each component. To further investigate the efficacy of each component, we replace each part with the normally used solutions to obtain three variants: (1) RoTTA w/o RBN, replace RBN with test-time BN in TENT [70]; (2) RoTTA w/o CSTU, directly adapt the model on test stream; (3) RoTTA w/o robust training (RT), directly adapt the model only with entropy minimization. As shown in Table 5, we can observe that significant performance degradation occurs for all variants, proving that every part of our proposed method is valid for PTTA. Take one component for a detailed example, without RBN robustly normalizing feature maps, the performance of RoTTA drops 50.2% and 16.3% on CIFAR10-C and CIFAR100-C respectively, proving that RBN is robust enough to tackle the problem of normalization of correlatively sampled data streams. CSTU enables RoTTA to adapt to a more stable distribution by maintaining a timely and confident snapshot of the test distribution. Meanwhile, robust training with timeliness greatly reduces the accumulation of errors. Every component behaves significantly to enable effective adaptation under PTTA.

Effect of the distribution changing order. To exclude the effect of a fixed order of distribution changing, we conducted experiments on ten different sequences of changes on CIFAR10-C and CIFAR100-C with independently and



(a) CIFAR10-C. (b) CIFAR100-C. (c) δ. (d) Batch size.

Figure 4. (a) & (b) we adapt the model continually to different corruptions of 10 different orders with independently and correlatively sampled test streams on CIFAR10-C and CFAR100-C respectively and report their average classification error. (c) & (d) we verify the effect of δ and batch size to different methods on CIFAR100-C respectively.

Table 5. Classification error of different variants of our RoTTA.

Variant	CIFAR10-C	CIFAR100-C	Avg.
RoTTA w/o RBN	75.4	51.3	63.4
RoTTA w/o CSTU	47.1	46.3	46.7
RoTTA w/o RT	78.2	95.0	81.6
RoTTA	**25.2**	**35.0**	**30.1**

correlatively sampled test streams respectively. As shown in Figure 4a and 4b, no matter what kind of setup, RoTTA can achieve excellent results. The detailed results on the correlatively sampled test streams are shown in Table 6, RoTTA achieves 4.3% and 4.7% progress on CIFAR10-C and CIFAR100-C respectively. This shows that RoTTA can adapt the model robustly and effectively in long-term scenarios where distribution continually changes and test streams are sampled either independently or correlatively, making it a good choice for model deployment.

Effect of Dirichlet concentration parameter δ. We vary the value of δ on CIFAR100-C and compare RoTTA with other approaches in Figure 4c. As the value of δ increases, the performance of BN [53], PL [39], TENT [70] and CoTTA [73] drops quickly, because they never consider the increasing correlation among test samples. NOTE [19] is stable to correlatively sampled test streams but does not consider the distribution changing, causing ineffective adaptation. Meanwhile, the higher correlation between test samples will make the propagation of labels more accurate, which is why the result of LAME [5] slightly improves. Finally, excellent and stable results once again prove the stability and effectiveness of RoTTA.

Effect of batch size. In real scenarios, considering deployment environments may use different test batch sizes, we conduct experiments with different values of test batch sizes and results are shown in Figure 4d. For a fair comparison, we control the frequency of updating the model of RoTTA so that the number of samples involved in back-propagation is the same. As the batch size increases, we can see that all of the compared methods have a significant improvement except for lame which has a slight decrease. This is because the number of categories in a batch increases with the

Table 6. Average classification error of tasks CIFAR10 → CIFAR10-C and CIFAR100 → CIFAR100-C while continually adapting to different corruptions of 10 different orders at the highest severity 5 with correlatively sampled test stream.

Method	CIFAR10-C	CIFAR100-C	Avg.
Source	43.5	46.4	46.9
BN [53]	75.2	52.9	64.1
PL [39]	75.2	52.9	60.1
TENT [70]	82.3	93.2	87.8
LAME [5]	39.5	40.6	40.1
NOTE [19]	30.5	76.1	53.3
CoTTA [73]	83.1	52.8	67.9
RoTTA	**26.2**(+4.3)	**35.9**(+4.7)	**31.1**(+9.0)

increasing batch size, causing the overall correlation to become lower but the propagation of labels to become more difficult. Most significantly, RoTTA achieves the best results across different batch sizes, demonstrating its robustness in dynamic scenarios once again.

5. Conclusion

This work proposes a more realistic TTA setting where distribution changing and correlative sampling occur simultaneously at the test phase, namely Practical Test-Time Adaptation (PTTA). To tackle the problems of PTTA, we propose **Ro**bust **T**est-**T**ime **A**daptation (RoTTA) method against the complex data stream. More specifically, a group of robust statistics for the normalization of feature maps is estimated by robust batch normalization. Meanwhile, a memory bank is adopted to capture a snapshot of the test distribution by category-balanced sampling with considering timeliness and uncertainty. Further, we develop a time-aware reweighting strategy with a teacher-student model to stabilize the adaptation process. Extensive experiments and ablation studies are conducted to verify the robustness and effectiveness of the proposed method. We believe this work will pave the way for thinking about adapting models into real-world applications by test-time adaptation algorithm.

Acknowledgements. This paper was supported by National Key R&D Program of China (No. 2021YFB3301503), and also supported by the National Natural Science Foundation of China under Grant No. 61902028.

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019. 3

[2] Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jörn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *WACV*, pages 2603–2612, 2022. 1, 3

[3] Mathilde Bateson, Herve Lombaert, and Ismail Ben Ayed. Test-time adaptation with shape moments for image segmentation. In *MICCAI*, pages 736–745, 2022. 1

[4] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *NeurIPS*, pages 2178–2186, 2011. 3

[5] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *CVPR*, pages 8344–8353, 2022. 2, 6, 7, 8

[6] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 3

[7] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, pages 295–305, 2022. 1, 4

[8] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, pages 3339–3348, 2018. 2

[9] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *CVPR*, pages 9137–9146, 2021. 3, 4

[10] Boris Chidlovskii, Stéphane Clinchant, and Gabriela Csurka. Domain adaptation in the absence of source domain data. In *KDD*, pages 451–460, 2016. 3

[11] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *ECCV*, pages 440–458, 2022. 1

[12] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. In *Neurips*, 2021. 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[14] Ying-Jun Du, Jun Xu, Huan Xiong, Qiang Qiu, Xiantong Zhen, Cees G. M. Snoek, and Ling Shao. Learning to learn with variational information bottleneck for domain generalization. In *ECCV*, pages 200–216, 2020. 3

[15] Sayna Ebrahimi, Sercan Ö. Arik, and Tomas Pfister. Test-time adaptation for visual document understanding. *CoRR*, abs/2206.07240, 2022. 1

[16] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. In *NeurIPS*, 2022. 1

[17] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 1, 2

[18] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N. Metaxas. Visual prompt tuning for test-time domain adaptation. *CoRR*, abs/2210.04831, 2022. 3

[19] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Robust continual test-time adaptation: Instance-aware BN and prediction-balanced memory. In *NeurIPS*, 2022. 1, 2, 3, 4, 6, 7, 8

[20] Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J Zico Kolter. Test time adaptation via conjugate pseudo-labels. In *NeurIPS*, 2022. 1, 3, 4

[21] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2004. 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 6

[23] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 2, 6

[24] Hengguan Huang, Xiangming Gu, Hao Wang, Chang Xiao, Hongfu Liu, and Ye Wang. Extrapolative continuous-time bayesian neural network for fast training-free test-time adaptation. In *NeurIPS*, 2022. 1

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 3, 4

[26] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, pages 2427–2440, 2021. 1, 3

[27] Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, pages 577–584, 2011. 3

[28] Minguk Jang and Sae-Young Chung. Test-time adaptation via self-training with nearest neighbor information. *CoRR*, abs/2207.10792, 2022. 3, 4

[29] Junho Kim, Inwoo Hwang, and Young Min Kim. Ev-tta: Test-time adaptation for event-based object recognition. In *CVPR*, pages 17724–17733, 2022. 1

[30] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[31] James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. 3

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1

[34] Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *ICML*, pages 5468–5479, 2020. 3

[35] Jogendra Nath Kundu, Naveen Venkat, Rahul M. V., and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, pages 4543–4552, 2020. 3

[36] 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. 3

[37] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022. 3

[38] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015. 1

[39] Dong-Hyun Lee et al. 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 896, 2013. 6, 7, 8

[40] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, pages 3490–3497, 2018. 1, 3

[41] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C. Kot. Domain generalization with adversarial feature learning. In *CVPR*, pages 5400–5409, 2018. 1, 3

[42] Shuang Li, Binhui Xie, Qiuxia Lin, Chi Harold Liu, Gao Huang, and Guoren Wang. Generalized domain conditioned adaptation network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4093–4109, 2022. 1

[43] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *CVPR*, pages 11516–11525, 2021. 2

[44] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2018. 3

[45] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, pages 6028–6039, 2020. 1, 3

[46] 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? In *NeurIPS*, pages 21808–21820, 2021. 3

[47] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *CVPR*, pages 1215–1224, 2021. 3

[48] Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Transferable representation learning with deep adaptation networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(12):3071–3085, 2019. 1, 2

[49] Wenao Ma, Cheng Chen, Shuang Zheng, Jing Qin, Huimao Zhang, and Qi Dou. Test-time adaptation with calibration of medical image classification nets for label distribution shift. In *MICCAI*, pages 313–323, 2022. 1

[50] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *ICML*, pages 7313–7324, 2021. 3

[51] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009. 2

[52] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18, 2013. 1, 3

[53] Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, abs/2006.10963, 2020. 4, 6, 7, 8

[54] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, pages 16888–16905, 2022. 1

[55] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, volume 162, pages 16888–16905, 2022. 4, 5, 6

[56] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. 1

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 6

[58] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 2, 6

[59] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. 2008. 1

[60] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017. 3

[61] Amelie Royer and Christoph H Lampert. Classifier adaptation at prediction time. In *CVPR*, pages 1401–1409, 2015. 3

[62] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018. 2

[63] Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. MM-TTA: multi-modal test-time adaptation for 3d semantic segmentation. In *CVPR*, pages 16907–16916, 2022. 1, 3

[64] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022. 1, 3

[65] 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, 2020. 1, 2, 3, 4

[66] Rishabh Tiwari, KrishnaTeja Killamsetty, Rishabh K. Iyer, and Pradeep Shenoy. GCR: gradient coreset based replay buffer selection for continual learning. In *CVPR*, pages 99–108, 2022. 3

[67] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, pages 7472–7481, 2018. 2

[68] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015. 2

[69] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017. 1

[70] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 1, 2, 3, 4, 6, 7, 8

[71] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Trans. Knowl. Data Eng.*, 2022. 1

[72] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 1

[73] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7191–7201, 2022. 1, 2, 3, 4, 6, 7, 8

[74] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Incremental adversarial domain adaptation for continually changing environments. In *ICRA*, pages 4489–4495, 2018. 3

[75] Binhui Xie, Shuang Li, Mingjia Li, Chi Harold Liu, Gao Huang, and Guoren Wang. Sepico: Semantic-guided pixel contrast for domain adaptive semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1–17, 2023. 2

[76] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 6

[77] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pages 1426–1435, 2019. 2

[78] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. In *ICLR*, 2021. 3

[79] Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *ICCV*, pages 8978–8987, 2021. 3

[80] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6

[81] Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022. 1, 4

[82] Yizhe Zhang, Shubhankar Borse, Hong Cai, and Fatih Porikli. Auxadapt: Stable and efficient test-time adaptation for temporally consistent video semantic segmentation. In *WACV*, pages 2633–2642, 2022. 1, 3

[83] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[84] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *ICLR*, 2021. 3

[85] 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