Distribution Alignment for Fully Test-Time Adaptation with Dynamic Online Data Streams

Ziqiang Wang[1], Zhixiang Chi[2], Yanan Wu[3], Li Gu[1], Zhi Liu[4]✉, Konstantinos Plataniotis[2]✉, and Yang Wang[1]✉

[1] Concordia University, Canada
{ziqiang.wang,li.gu}@mail.concordia.ca, yang.wang@concordia.ca
[2] University of Toronto, Canada
zhixiang.chi@mail.utoronto.ca, kostas@ece.utoronto.ca
[3] Beijing Jiaotong University, China ynwu0510@bjtu.edu.cn
[4] Shanghai University, China liuzhisjtu@163.com

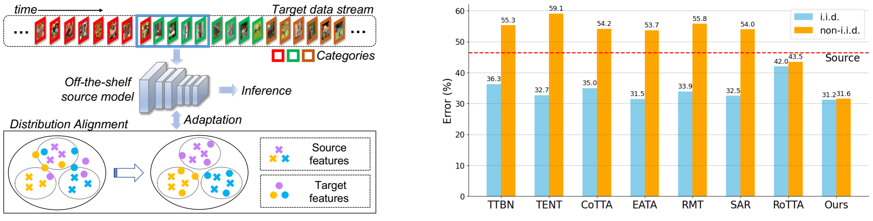
Abstract. Given a model trained on source data, Test-Time Adaptation (TTA) enables adaptation and inference in test data streams with domain shifts from the source. Current methods predominantly optimize the model for each incoming test data batch using self-training loss. While these methods yield commendable results in ideal test data streams, where batches are independently and identically sampled from the target distribution, they falter under more practical test data streams that are not independent and identically distributed (non-i.i.d.). The data batches in a non-i.i.d. stream display prominent label shifts relative to each other. It leads to conflicting optimization objectives among batches during the TTA process. Given the inherent risks of adapting the source model to unpredictable test-time distributions, we reverse the adaptation process and propose a novel Distribution Alignment loss for TTA. This loss guides the distributions of test-time features back towards the source distributions, which ensures compatibility with the well-trained source model and eliminates the pitfalls associated with conflicting optimization objectives. Moreover, we devise a domain shift detection mechanism to extend the success of our proposed TTA method in the continual domain shift scenarios. Our extensive experiments validate the logic and efficacy of our method. On six benchmark datasets, we surpass existing methods in non-i.i.d. scenarios and maintain competitive performance under the ideal i.i.d. assumption.

Keywords: Test-time adaptation · Domain shift · Label shift

1 Introduction

The unprecedented success of deep models [7, 18, 62, 67] is conditioned on the assumption that the training and test data are drawn from the same distribution [55]. However, such an assumption is delicate in ever-changing deployment environments [13, 25], leading to domain shift and performance deterioration.

✉ Corresponding authors. Our code is available at github.com/WZq975/DA-TTA.



(a) **(b)**

Fig. 1: (a) Overview of our method for fully TTA. With each time step, there are distribution shifts (label shifts). Our proposed method aligns the distributions of test-time features with those of the source, not only mitigating the domain shift, but ensuring robust TTA in the non-i.i.d. data streams. (b) Average classification errors of TTA methods from CIFAR100 (source) to CIFAR100-C (target) under the fully TTA with i.i.d and non-i.i.d data stream settings. Lower is better. The red "Source" line indicates the source model's result without adaptation. Label shifts in the non-i.i.d. test data stream degrade TTBN and self-training based methods (SAR and RoTTA are designed for non-i.i.d. data streams).

Test-Time Adaptation (TTA) is a line of research that mitigates domain shift by continually adapting to the unlabeled data stream in a target domain before inference [6, 8, 29, 53, 55]. There are two main categories of TTA: 1) Test-Time Training (TTT) [53], where customized model training (*e.g.*, adding auxiliary tasks [32,59]) is performed offline using the source data and performs adaptation on test data. 2) Fully TTA [55], adapts an off-the-shelf model without altering offline training. Our work focuses on fully TTA which poses a greater challenge as minimal training is allowed.

There have been a series of studies on fully TTA to tackle the challenges of learning on unlabeled data. One notable method is Test-time Batch Normalization (TTBN) [39, 48, 60]. TTBN adjusts the BN layers, allowing them to normalize the feature leveraging the *batch statistics from the current test data batch*, rather than the population statistics from the training phase. Given its effectiveness, TTBN has become a cornerstone for recent TTA research. Following this, there has been a surge in self-training-based TTA methods, primarily hinging on two adaptation objectives. The first employs entropy minimization (EM) [16, 40, 50, 55, 71, 73], pushing the model to make predictions with low entropy for the incoming test data. This ensures that the inference is well-distanced from the classification boundary [49], thereby enhancing model performance on the test data. The second utilizes a teacher-student self-training framework [11, 57, 69, 74]. Here, a teacher model assigns pseudo labels to the student, allowing the latter to be trained in a manner similar to supervised learning.

Both TTBN and self-training-based TTA methods aim to tailor the model *towards the incoming unknown test data batch*. Their notable performances have been observed in ideal circumstances where every test data batch is independent and identically distributed (i.i.d., balanced classes) from the target domain. In the online fashion, as the model keeps updating, the i.i.d scenario ensures stable optimization from batch to batch. However, real-world scenarios are rarely so accommodating. In applications like self-driving vehicles and robot-vision systems, the batches of image feed are temporally correlated, making it non-i.i.d (imbalanced classes) [16,17]. In the non-i.i.d. data streams, the long-tailed problem may occur, where a minority of classes dominate the current batch [72] (see Fig. 1a). As each incoming batch of data contains different sets of dominated classes, the distributions among batches are diverse, leading to conflicting optimization. As depicted in Fig. 1b, for TTBN, the adaptation fails because test data batches in long-tailed distributions do not provide true target domain statistics for the BN layers. For self-training-based methods, the varying long-tailed target distributions in TTA sessions lead to conflicting optimization objectives [26], which severely impair model performance, and may even cause it to collapse [63,64].

To address the aforementioned challenges, we propose a surprisingly simple yet effective method for robust TTA, as illustrated in Fig. 1a. Rather than adapting the model to unpredictable test-time distributions, we reverse such a process and propose to set the source feature distribution as a reference and pull test data towards it. As a result, conflicting optimization objectives among data batches can be alleviated. Specifically, we propose to optimize the source model's affine layers using a Distribution Alignment (DA) loss. This loss minimizes the divergence between test feature distributions and the source distributions, thereby ensuring the test data's features align with the source model for compatibility. Furthermore, to accommodate scenarios featuring continuous domain shifts in test data streams, namely continual TTA [11,57,69], we devise a domain shift detection mechanism that tracks changes in feature distributions. It improves our TTA method's efficacy in continuous domain environments. As demonstrated in Fig. 1b, our method outshines others in handling both i.i.d. and non-i.i.d. data streams, effectively navigating the challenges associated with non-i.i.d. streams.

Main Contributions: (1) Our distribution alignment loss addresses the TTA challenges in non-i.i.d. scenarios by aligning test features with source distributions, ensuring they mesh with the source model and preventing degradation from conflicting optimization objectives. (2) We propose a domain shift detection mechanism that tracks feature distributions, enhancing our TTA method's performance for continual TTA in non-i.i.d. data streams. (3) Our method surpasses recent state-of-the-art methods (*e.g.*, $\sim 6\%$ on ImageNet-C/CIFAR100-C) across six datasets with different types of domain shifts in non-i.i.d. scenarios, while maintaining comparable performance under i.i.d. assumption.

2 Related Work

Unsupervised Domain Adaptation (UDA). UDA addresses the distribution shift by jointly training on the labeled source and unlabeled target data [4,27–29,58]. One popular approach is to learn domain-invariant features by minimizing a certain measure of divergence between the source and target distributions (*e.g.* [24,31,35,43,52]). Another line of studies involves embedding a "domain discriminator" within the network, which is applied to develop indistinguishable feature space (*e.g.* [14,36,42,45]). However, the necessity of having access to both source and target domains during training limits the usability of these methods.

Source-free Domain Adaptation (SFDA). SFDA aims to adapt source models to unlabeled target domains without accessing the source domain data [1,22, 54, 56, 66]. Among these, SHOT [29] suggests learning target-specific features through information maximization and pseudo-label prediction. SFDA-DE [10] works on domain alignment by estimating source class-conditioned feature distribution and minimizing a contrastive adaptation loss. DSiT [47] utilizes the queries of a vision transformer to induce domain-specificity and train the unified model to enable a disentanglement of task- and domain-specificity. Most existing source-free methods [26,56,65] operate offline and require an analysis of the entire test dataset, along with several adaptation epochs for model updates. Specially, BUFR [12] pre-computes and stores marginal distributions for each feature on source data using a soft binning function. It then realizes adaptation by restoring the test features with the stored marginal distributions. The philosophy of BUFR can be related to our work, thus, we also include a comparison with this SFDA method in our experiments.

Test-time Adaptation (TTA). TTA can be categorized into Test-Time Training [53] (TTT) and Fully TTA [55], differentiated by the presence of prior joint training. TTT leverages both supervised and self-supervised losses to train a source model, which is then fine-tuned during TTA using self-supervised learning [2,34]. Fully TTA, in contrast, performs inference and adaptation directly in test data streams without prior training. A notable method in this setting is Test-Time Batch Normalization (TTBN) [39,48], which utilizes test-time batch statistics within BN layers for adaptation. Subsequently, optimization-based methods, including entropy minimization [16, 40, 50, 55, 71, 73] and teacher-student self-training [11,37,38,57,69], have been developed. EATA [40] alleviates redundant optimization in test streams by employing a mechanism that identifies redundant samples. It tracks model outputs and skips the optimization for samples that are similar to previous ones. Our domain shift detection mechanism also adopts a tracking philosophy, albeit with a different focus and objective: to monitor feature distribution and detect domain shifts. Besides, LAME [3] focuses on adjusting output assignments rather than tuning parameters for TTA, while ODS [75] optimizes the estimation of label distribution to enhance self-training-based TTA methods in scenarios involving label shift. Moreover, DDA [15] employs diffusion models to align target images with the source domain, then realizes classification without adapting the source model.

Modified TTBN for TTA in Non-i.i.d. Streams. TTBN [39,48] establishes a strong baseline for TTA, yet it encounters difficulties in non-i.i.d. data streams or when dealing with small batch sizes. This is because incoming batches are class-imbalanced and provide biased statistics for BN layers. Subsequent works have modified TTBN to better handle non-i.i.d. streams or limited batch sizes. MEMO [71] and TTN [30] combine source population statistics with dynamic test batch statistics, while DELTA [73] applies a moving average of test batch

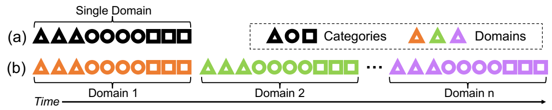


Fig. 2: (a) Non-i.i.d. data streams. (b) Continual TTA with non-i.i.d. data streams.

statistics for batch normalization. Furthermore, NOTE [16] adjusts the normalization layers of TTBN by selectively incorporating instance normalization. In addition, both NOTE [16] and RoTTA [69] employ a resampling memory bank that collects and stores test samples from different estimated classes and updates test batch statistics from the stored samples using a moving average.

Overall, prior works modify the computing of batch normalization, trying to stabilize the normalization process for the incoming test batch. Differently, this work investigates the correlation between model accuracy and the change in intermediate feature distribution due to imbalanced or balanced classes (Sec. 3.2). And we introduce our Distribution Alignment method, which directly optimizes the distribution of features for all test batches towards the same source reference.

3 Methodology

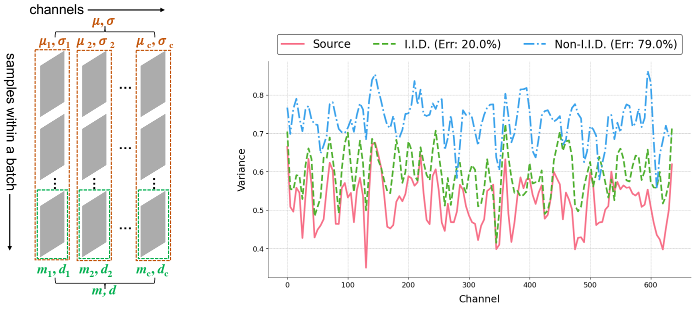
3.1 Problem Definition

Fully TTA [55] encompasses a scenario where a model, pre-trained on a labeled source dataset $\{(x, y) \sim P_S(x, y)\}$, is subjected to a stream of unlabeled test data from a target dataset $\{(x, y) \sim P_T(x, y)\}$. This target dataset presents a domain shift from the source, indicated by $P_S(x) \neq P_T(x)$ and $P_S(y|x) = P_T(y|x)$ [51]. After deployment, the model updates itself based on the current data it receives, without using the source data. In pioneer work, fully TTA assumes that the distributions of target data over time, $P_T(x, y \mid t)$, are i.i.d. that is consistent with $P_S(x, y)$. However, our focus is on practical scenarios where $P_T(x, y \mid t)$ is **non-i.i.d.** and changes over time. Therefore, **fully TTA in non-i.i.d. data streams** demands the management of both the domain shift from source to target and the distribution shifts (label shifts) that occur at each time step. Besides, we also consider **continual TTA** [11,50,69]. This setting extends the fully TTA from a single target domain to a sequence of continuously shifting target domains: $P_{T_1}(x), P_{T_2}(x), \ldots, P_{T_n}(x)$, as depicted in Fig. 2.

3.2 Motivation on TTA in Dynamic Online Settings

TTBN [39,48] sets a strong baseline underpinning a series of TTA works [11,30, 37,38,40,55,57]. We first analyze why TTBN experiences performance drop in dynamic online data streams follow by our motivation.

Analysis of TTBN in Non-i.i.d. Data Streams. Fig. 1b shows the commendable efficacy of TTBN [39,48] in i.i.d. test data streams, positioning it



(a) **(b)**

Fig. 3: **(a)** For clarity, we use $\boldsymbol{\mu}, \boldsymbol{\sigma}$ to denote the batch statistics, while using $\boldsymbol{m}, \boldsymbol{d}$ to denote the mean and standard deviation of the feature distribution for each sample. **(b)** Visualization of the average statistics $(\overline{\boldsymbol{d}^2}(\hat{\mathbf{X}}^{(i)}))$ of feature distributions in a BN layer of the source model through TTBN method. We feed both source and target data streams (both i.i.d. and non-i.i.d.) and record the average variance for each channel in the corresponding BN layer. For clarity, the curves are downsampled by a factor of 5 and smoothed. The figure shows that the distribution statistics for the i.i.d. stream are nearing those of the source data, associated with a 20% error. In contrast, the statistics for the non-i.i.d. stream notably deviate from the source, associated with a 79% error.

as a strong baseline method. However, its performance wanes when exposed to non-i.i.d. data streams. We attribute this degradation to misleading distribution statistics provided by non-i.i.d data batches. Specifically, TTBN's adjustment to the source model lies on the test-time statistics—the means $\boldsymbol{\mu}$ and standard deviations (stds) $\boldsymbol{\sigma}$ of each Batch Normalization layer:

$$\boldsymbol{\mu} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{X}^{(i)}, \boldsymbol{\sigma}^2 = \frac{1}{b}\sum_{i=1}^{b}(\mathbf{X}^{(i)} - \boldsymbol{\mu})^2, \tag{1}$$

where $\mathbf{X}^{(i)}$ is the input feature corresponding to i^{th} sample in a batch with batch size b. The $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ enable an affine transformation that normalizes the feature $\mathbf{X}^{(i)}, i \in [1, b]$ to match preferred distributions of the well-trained source model:

$$\hat{\mathbf{X}}^{(i)} = \frac{\mathbf{X}^{(i)} - \boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}} = \frac{\mathbf{X}^{(i)}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}} + \frac{-\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2 + \epsilon}}, \tag{2}$$

$$\Rightarrow \begin{cases} \boldsymbol{m}(\hat{\mathbf{X}}^{(i)}) & = \frac{1}{\sqrt{\boldsymbol{\sigma}^2+\epsilon}} \cdot \boldsymbol{m}(\mathbf{X}^{(i)}) + \frac{-\boldsymbol{\mu}}{\sqrt{\boldsymbol{\sigma}^2+\epsilon}}, \\ \boldsymbol{d}(\hat{\mathbf{X}}^{(i)}) & = \frac{1}{\sqrt{\boldsymbol{\sigma}^2+\epsilon}} \cdot \boldsymbol{d}(\mathbf{X}^{(i)}), \end{cases} \tag{3}$$

where $\hat{\mathbf{X}}^{(i)}$ is transformed from $\mathbf{X}^{(i)}$, and $\boldsymbol{m}(\hat{\mathbf{X}}^{(i)})$, $\boldsymbol{d}^2(\hat{\mathbf{X}}^{(i)})$ are the mean and variance of each transformed feature map. For clarity, we use $\boldsymbol{\mu}, \boldsymbol{\sigma}$ to denote the *batch* statistics, while using $\boldsymbol{m}, \boldsymbol{d}$ to denote the mean and standard deviation of the feature distribution for a *single* sample, as illustrated in Fig. 3a.

Implications for TTA. During test-time, when fed a data batch, the TTBN layers transform their features, trying to approach $\hat{\mathbf{X}}^{(i)}$ towards the distributions of the source data features. The domain shift can be effectively mitigated and performance can be preserved if the transformed features approximate the source distribution. Fig. 3b provides a visual exploration of the impact of non-i.i.d. data streams on the transformed feature distribution. With a frozen source model, three different data streams are assessed through the TTBN method: an i.i.d. source data stream, and both i.i.d. and non-i.i.d. target data streams. These visualizations randomly spotlight one of BN layers, with the x-axis denoting feature channels, and the y-axis portraying the average variances $\overline{\boldsymbol{d}^2}(\hat{\mathbf{X}}^{(i)})$. The means are approximating zeros that are omitted here.

The key takeaways include: (a) I.i.d. target data has its transformed feature distributions closer to the source distributions, and **mild performance drop** is observed (i.e., error rate = 20%). (b) Conversely, non-i.i.d. streams manifest feature distributions that deviate from the source, correlating with the observed **significant performance dip** (i.e., error rate = 79%). Therefore, the performance of TTBN method is greatly hampered under the non-i.i.d. (dynamic online) setting due to the drifting of target feature distributions. On the other hand, as the distribution (label) for each non-i.i.d. batch differs, it causes the gradient conflict [68] among batches when the model is updating towards test data. The performance is further impeded due to such conflicting optimizations.

To this end, we aim to narrow the disparity in distributions between non-i.i.d. test features and source features by steering the feature distributions **back to the source**, ensuring they are aptly managed by the source model. Moreover, as the source distribution is set as the *"reference"* for all test data streams, it sidesteps conflicting optimizations on distinctively distributed batches, thus preventing degradation of the well-trained source model.

3.3 Distribution Alignment for TTA

We propose the Distribution Alignment (DA) loss, a simple yet effective method to provide consistent optimization objectives. It avoids the instability caused by conflicting objectives, and effectively counteracts domain shifts by steering the test-time feature distributions towards the source domain. The DA loss is applied to the features from intermediate layers (DA is applied to multiple layers, we omit the layer notation here for simplicity) of the source model. Upon processing a batch of input data, we calculate distribution statistics of the features in the model. For one of the intermediate features, $\mathbf{X}$, we have:

$$\boldsymbol{m}_j = \frac{1}{H \cdot W} \sum_{p=1}^{H \cdot W} \mathbf{X}_{j,p}, \qquad \boldsymbol{d}^2{}_j = \frac{1}{H \cdot W} \sum_{p=1}^{H \cdot W} (\mathbf{X}_{j,p} - \boldsymbol{m}_j)^2, \tag{4}$$

where $j \in \{1, \ldots, C\}$, C is the channel, and $H \cdot W$ represents the spatial size of each feature map. It is important to note that prior to TTA, we pre-compute the average feature distribution statistics $\overline{\boldsymbol{m}}_j^S, \overline{\boldsymbol{d}^2}_j^S$ of source data offline. This

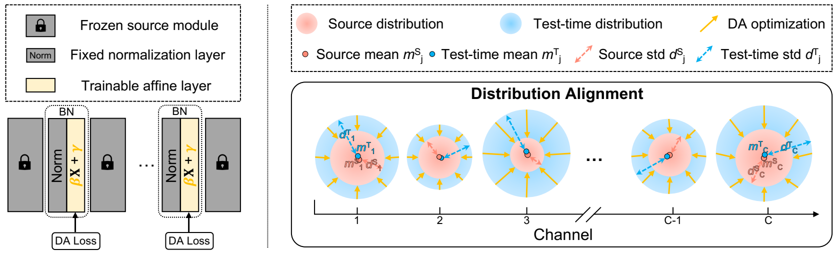


Fig. 4: Illustration of the distribution alignment loss applied to features post-BN layers. This loss optimizes the affine parameters of BN layers to transform the feature distributions towards the pre-computed source distributions. Each circle represents the distribution of a channel in the feature following an affine layer, characterized by a mean (center) and a standard deviation (radius).

computation is performed only once, after which the source statistics are retained for ongoing use. This kind of operation before deployment time is also adopted in other TTA methods, such as EATA [40] and RMT [11]. During TTA, when a batch of test data is fed into the model, the test statistics $\boldsymbol{m}_j^T, \boldsymbol{d^2}_j^T$ are also computed using Eq. (4). Subsequently, the DA loss is computed based as:

$$\mathcal{L}_{DA} = \frac{1}{C}\sum_{j=1}^{C}\left(\left|\boldsymbol{m}_j^T - \overline{\boldsymbol{m}}_j^S\right| + \left|\boldsymbol{d^2}_j^T - \overline{\boldsymbol{d^2}}_j^S\right|\right), \tag{5}$$

where $|\cdot|$ denotes the absolute value operation. Therefore, the DA loss quantifies the disparity between the feature distributions of the source and test data, with the objective of pulling test-time feature distributions back to the source domain through optimization as shown in the right side of Fig. 4.

Feature distribution (means and variances) can be linearly manipulated via affine transformations. Hence, we utilize the affine layers in BN layers to be optimized by the DA loss, as depicted on the left side of Fig. 4. Alternative strategies for the selection of affine layers, such as integrating external affine layers instead of utilizing those within BN layers, are discussed in Appendix G.

At inference, BN layers utilize the population mean $\boldsymbol{\mu}_{popu}$ and variances $\boldsymbol{\sigma}^2_{popu}$ computed at pre-training for normalization. Before the TTA process starts, we update the such statistics in [23] towards the first batch of test data as:

$$\boldsymbol{\mu}_{norm} = \alpha \cdot \boldsymbol{\mu}_{popu} + (1-\alpha)\cdot \boldsymbol{\mu}_{B_1}, \tag{6}$$

$$\boldsymbol{\sigma}^2_{norm} = \alpha \cdot \boldsymbol{\sigma}^2_{popu} + (1-\alpha)\cdot \boldsymbol{\sigma}^2_{B_1}, \tag{7}$$

where $\boldsymbol{\mu}_{B_1}, \boldsymbol{\sigma}^2_{B_1}$ are the statistics of the first batch from the test data stream, and α is a hyper-parameter. This modification offers a more favorable starting point for optimization, ensuring that the initial distribution discrepancy between the test and source features is not excessively large.

Additionally, we explore the synergistic effect of combining the DA loss with the entropy minimization (EM) loss:

$$\mathcal{L}_{EM} = \sum_{m} \left[\mathbb{1} \left(\max_{n} \hat{y}_n > \theta \right) \sum_{n=1}^{N} -p(\hat{y}_n) \log p(\hat{y}_n) \right], \tag{8}$$

$$\mathcal{L}_{final} = \mathcal{L}_{DA} + \mathcal{L}_{EM}, \tag{9}$$

where $\hat{y}_n$ denotes the predicted probability for class n, with n ranging from 1 to N, $\mathbb{1}(\cdot)$ denotes an indicator function, θ is the confidence threshold, and m is the batch size. Sec. 4.4 will reveal that, additional EM loss further improves, although DA loss alone can achieve SoTA performance.

3.4 Domain Shift Detection in Continual TTA Setting

In certain application scenarios, it is essential for deployed models to automatically process data streams with continual domains without manual intervention [5, 33], as shown in Fig. 2b. The DA loss is designed to pull the feature distributions in current test domain to the source distributions. When a new target domain is encountered, the model, whose affine layers are tailored to the last tested domain, may apply unsuitable affine transformations on the features of the new domain. This is particularly problematic in the event of significant domain shifts, as the discrepancy between the test-time feature distributions and the source distributions can increase substantially, thereby raising the risk of convergence to a suboptimal local minimum.

To improve the performance of our method in the continual TTA setting [11, 50, 69], we introduce a domain shift detection mechanism. This mechanism tracks the DA loss $\mathcal{L}_{DA}$, which reflects the discrepancy between test-time feature distributions and source distributions. A domain shift is detected if the average discrepancy within a short-term window is larger than the average discrepancy within a long-term window by a predefined margin, $\frac{\sum_{i=0}^{p} \mathcal{L}_{DA}^{B_{t-i}}}{p} > \tau \cdot \frac{\sum_{i=0}^{q} \mathcal{L}_{DA}^{B_{t-i}}}{q}$, where $\mathcal{L}_{DA}^{B_t}$ denotes the DA loss of the current batch, p, q denote the lengths of short-term and long-term windows, and the τ is the threshold factor. Upon detecting a new domain, the model's trainable affine layers are reset to their initial states and the normalization layers in BN layers are reset according to Eq. (6) and Eq. (7). For more details on the domain shift detection mechanism, see Algorithm 1, Appendix A. It is noteworthy that in the continual TTA setting, we employ both distribution alignment and domain shift detection, whereas for the fully TTA setting, we exclusively utilize distribution alignment.

4 Experiments

We conduct comparative experiments on TTA benchmarks with state-of-the-art TTA methods: TTBN [39, 48], TENT [55], MEMO [71], LAME [3], CoTTA [57], EATA [40], NOTE [16], RoTTA [69], RMT [11], DELTA [73], and SAR [41]. We also compare our method with a related SFDA work, BUFR [12], in Appendix F.1. For a fair comparison, we adopt the codebase from RMT [11] which integrates many SoTA TTA methods. In Appendix F.1, we also integrate our method into the official NOTE [16] codebase for a direct comparison with NOTE [16].

4.1 Datasets

CIFAR10/100-C, ImageNet-C. We conduct experiments on the CIFAR10-C, CIFAR100-C, and ImageNet-C [20] that are common TTA benchmarks. They have 15 types of corruptions (target domains) applied on test and validation. Each type of corruption has 5 severity levels, of which we use the highest. On each target domain, CIFAR10-C/CIFAR100-C/ImageNet-C has 10 000/10 000/50 000 images and 10/100/1000 classes.

ImageNet-R, ImageNet-D, ImageNet-A. We additionally conduct experiments on other types of domain shifts. ImageNet-R [19] contains 200 ImageNet classes with different textures and styles. ImageNet-D [46], re-proposed from DomainNet [44], maps classes to those in ImageNet, removing unmappable classes. Furthermore, ImageNet-A [21] comprises adversarially filtered images from 200 ImageNet classes. We use these datasets as target domains, with ImageNet serving as the source domain. Appendix B shows more details.

4.2 Implementation and Setup.

We evaluate our method in both fully TTA [16, 40, 41, 55, 73] and continual TTA [11, 50, 69] settings within non-i.i.d. scenarios. In the fully TTA setting, an off-the-shelf source model is online adapted to a data stream from a single target domain. In the continual TTA setting, the model is online adapted to a data stream that comprises a succession of domains, each fed sequentially one after the other. Following most existing TTA work, we use a pre-trained WideResNet-28 [70], ResNeXt-29 [61], and ResNet-50 [18] from the Robust-Bench benchmark [9] as source models for the CIFAR10-C, CIFAR100-C, and ImageNet-C/D/R, respectively, in experiments for all methods. In comparative experiments, non-i.i.d. data streams of CIFAR10/100-C and ImageNet-R/D/A are generated based on Dirichlet distribution with Dirichlet parameter δ set to 0.1, which controls the degree of temporal correlation of class labels in data streams. We provide illustration of non-i.i.d. data streams with different δ in Appendix C.1. For ImageNet-C, due to the low ratio of samples per class to the number of classes, we construct non-i.i.d. data streams by sorting the images according to their labels. More implementation details on ImageNet-D/R/A and hyper-parameter details can be found in Appendix C.2 and Appendix D.

4.3 Main Results

We conduct the comparative experiments in both the fully TTA (Tab. 1, Tab. 2, Tab. 3) and the continual TTA (Tab. 4) settings as described in Sec. 3.1.

Table 1: Fully test-time adaptation in non-i.i.d. test data streams. Classification error rates (%) are reported for each of the 15 target domains within the CIFAR-10-C, CIFAR-100-C, and ImageNet-C datasets, respectively.

	Method	bri	gla	jpe	con	def	imp	mot	sno	zoo	fro	pix	gau	ela	sho	fog	Mean
CIFAR10-C	Source	9.3	54.3	30.3	46.7	46.9	72.9	34.8	25.1	42.0	41.3	58.5	72.3	26.6	65.7	26.0	43.5
	CoTTA [57]	75.3	81.5	78.6	80.3	77.2	80.1	77.9	77.4	77.1	76.4	77.3	78.3	78.8	77.3	76.3	78.0
	RMT [11]	74.3	81.1	79.6	75.2	75.9	80.9	76.0	77.0	76.0	76.2	77.5	79.2	79.1	78.0	75.6	77.5
	TENT [55]	72.5	81.4	80.1	76.4	75.2	81.1	75.1	76.4	75.8	75.0	77.2	78.3	79.0	78.2	73.6	77.0
	TTBN [39, 48]	71.6	80.3	78.7	73.2	73.6	80.0	73.9	75.3	73.9	73.9	76.4	77.5	77.4	77.0	73.1	75.7
	SAR [41]	71.8	80.2	78.5	73.4	73.4	79.9	73.9	75.3	73.8	73.9	76.3	77.6	77.3	76.9	72.9	75.7
	EATA [40]	75.3	81.5	78.6	80.3	77.2	80.1	77.9	77.4	77.1	76.4	77.3	78.3	78.8	77.3	76.3	75.7
	MEMO [71]	8.3	48.7	27.5	30.5	32.6	56.5	27.7	20.1	29.4	29.4	51.3	57.2	25.2	50.6	20.6	34.4
	LAME [3]	**4.6**	42.9	**6.7**	40.8	25.8	62.7	**12.8**	**9.4**	**14.1**	25.5	53.3	77.9	**5.9**	67.3	**11.5**	30.7
	NOTE [16]	13.4	46.3	44.1	**9.9**	24.9	45.3	20.4	22.6	25.1	23.5	36.2	36.9	36.5	34.3	21.8	29.4
	RoTTA [69]	11.0	45.0	35.3	21.1	18.8	46.5	19.6	23.8	17.2	23.4	27.3	37.6	31.6	35.5	21.1	27.6
	DA-TTA (ours)	11.7	**42.1**	32.0	14.4	**17.8**	**40.7**	18.2	20.6	16.2	**20.6**	**24.4**	**31.0**	27.6	**28.8**	18.7	**24.3**
CIFAR100-C	Source	29.5	54.1	41.2	55.1	29.3	39.4	30.8	39.5	28.8	45.8	74.7	73.0	37.2	68.0	50.3	46.5
	LAME [3]	52.7	77.3	68.5	84.3	47.4	46.2	48.4	68.5	48.2	73.4	94.9	87.8	65.3	85.8	75.0	68.3
	TENT [55]	52.5	65.4	63.6	59.6	52.7	59.3	54.7	57.2	52.8	62.7	57.1	64.2	61.0	62.0	62.0	59.1
	RMT [11]	50.3	60.1	59.2	53.7	51.4	59.1	53.3	56.1	51.7	56.1	53.8	59.7	56.9	58.7	57.4	55.8
	TTBN [39, 48]	48.8	60.4	59.5	51.0	49.4	60.2	50.4	55.6	49.8	54.8	53.6	60.2	55.6	59.1	60.4	55.3
	NOTE [16]	43.2	64.9	60.9	38.7	45.5	62.2	46.4	51.9	46.5	52.0	54.3	65.6	58.5	64.1	62.5	54.5
	CoTTA [57]	48.7	57.8	55.9	54.1	50.2	56.9	51.0	54.8	49.8	53.7	51.0	57.2	54.9	56.3	60.0	54.2
	SAR [41]	48.6	59.1	58.3	51.3	48.8	54.7	50.2	54.5	49.1	54.4	51.9	58.3	55.3	56.6	58.0	54.0
	EATA [40]	48.6	58.6	57.8	51.2	49.2	53.8	50.3	53.7	49.7	54.3	51.9	57.9	55.6	56.3	56.8	53.7
	RoTTA [69]	30.6	48.1	48.2	61.9	32.1	49.9	34.6	40.6	32.8	48.2	40.7	49.2	41.6	49.2	45.3	43.5
	MEMO [71]	26.4	45.6	39.2	38.5	28.7	36.1	29.4	34.1	29.2	35.9	52.2	57.6	36.8	53.5	46.5	39.3
	DA-TTA (ours)	**23.8**	**37.8**	**37.3**	**26.6**	**25.3**	**33.5**	**27.2**	**31.5**	**25.2**	**31.4**	**29.8**	**38.5**	**32.8**	**37.3**	**36.0**	**31.6**
ImageNet-C	Source	41.1	90.2	68.4	94.6	82.1	98.2	85.2	83.1	77.5	76.7	79.4	97.8	83.0	97.1	75.6	82.0
	EATA [40]	95.1	99.7	98.5	99.6	99.5	99.5	99.1	98.5	98.5	98.6	98.5	99.5	98.2	99.4	97.9	98.7
	TENT [55]	92.7	98.5	95.5	99.2	98.2	97.8	98.0	97.0	96.0	97.0	95.1	97.8	96.0	97.8	94.2	96.7
	RMT [11]	93.5	98.3	95.7	97.6	98.1	97.4	96.9	96.1	96.2	96.1	95.2	97.4	95.9	97.6	94.6	96.4
	SAR [41]	91.9	98.2	94.6	98.9	98.1	97.6	97.8	96.3	95.7	96.7	94.6	97.8	95.3	97.9	93.6	96.3
	CoTTA [57]	91.8	98.2	94.9	98.0	98.1	97.2	96.7	95.3	95.6	95.0	94.4	97.3	95.2	97.1	93.7	95.9
	TTBN [39, 48]	91.1	98.1	94.8	97.4	98.0	97.5	96.6	95.1	95.4	94.8	94.3	97.5	95.1	97.4	93.6	95.8
	NOTE [16]	66.0	95.7	91.4	93.5	97.1	95.6	89.5	85.3	89.3	84.6	90.5	96.0	83.3	96.1	79.5	88.9
	DELTA [73]	50.9	90.2	69.2	97.5	90.4	86.5	87.2	78.7	74.2	78.2	64.7	86.6	68.2	85.8	64.0	78.1
	MEMO [71]	39.6	86.4	63.8	87.2	81.4	91.3	77.5	71.7	72.5	69.5	65.0	92.5	72.8	91.3	65.4	75.2
	RoTTA [69]	37.8	86.6	63.2	83.2	86.4	84.8	76.6	67.9	64.7	73.3	55.5	85.8	58.3	85.7	54.6	71.0
	LAME [3]	**27.6**	83.3	**45.6**	90.2	**57.3**	98.9	71.6	75.7	**60.2**	**52.6**	62.6	98.8	78.4	97.8	60.5	70.7
	DA-TTA (ours)	35.2	**77.3**	55.2	**70.8**	75.5	**78.0**	**69.5**	**63.9**	60.8	74.9	**50.3**	**79.7**	**53.4**	**78.6**	**49.7**	**64.8**

Fully TTA in Non-I.I.D. Data Streams. Tab. 1 presents the results of our method in comparison to other TTA methods on the commonly used corruption benchmark. Observing the "Mean" column reveals that over half of the prior methods yield results inferior to the source model without adaptation, suggesting an adaptation failure. Our method, denoted DA-TTA, outperforms competing methods across all the datasets, showcasing accuracy improvements of $3.3\%, 7.7\%, 5.9\%$ over the next best-performing methods, respectively. Furthermore, DA-TTA demonstrates robust performance across all target domains. In contrast, LAME excels in certain domains but significantly underperforms or even regresses relative to the "Source" in others. Besides, it is observed that many previous methods exhibit notably poorer results in non-i.i.d. streams compared to i.i.d. streams (shown in Appendix F.2). Our method, however, obtains close results in the i.i.d. and non-i.i.d. data streams.

Apart from the corruption domain shifts, Tab. 2 presents the results of evaluations on realistic domain shifts using the ImageNet-D and ImageNet-R datasets. Previous methods, except for RoTTA, fail to adapt in these non-i.i.d. streams (performing worse than the "Source"), while our method achieves an improvement of 3.9% on ImageNet-D and 5.3% on ImageNet-R compared to the "Source".

Table 2: Fully test-time adaptation in non-i.i.d. streams. Presented are error rates (%) on ImageNet-D and ImageNet-R datasets.

Method	ImageNet-D						ImageNet-R
	cli	inf	pai	real	ske	Mean	
Source	76.0	89.7	65.1	40.2	82.0	70.6	63.8
LAME [3]	88.0	99.7	70.5	**35.4**	92.6	77.2	84.8
TTBN [39, 48]	81.4	91.3	79.0	74.5	87.3	82.7	69.2
TENT [55]	80.9	90.8	79.2	76.6	87.6	83.0	68.3
CoTTA [50]	80.0	90.6	77.5	73.0	85.8	81.4	67.9
SAR [41]	80.3	90.0	78.5	74.9	86.0	81.9	67.6
EATA [40]	79.6	89.4	78.1	75.5	86.2	81.8	66.3
RoTTA [69]	70.3	86.0	63.3	40.7	77.6	67.6	61.5
DA-TTA(ours)	**69.2**	**85.4**	**61.5**	40.7	**75.9**	**66.5**	**58.5**

Table 3: Fully test-time adaptation. Error rates (%) on ImageNet-A dataset.

Method	i.i.d.	non-i.i.d.
Source	90.5	90.5
LAME [3]	95.2	95.3
TTBN [39, 48]	92.8	93.0
RMT [11]	95.5	95.5
TENT [55]	92.5	92.6
CoTTA [50]	93.0	93.2
SAR [41]	93.5	93.7
EATA [40]	92.8	92.7
RoTTA [69]	93.5	93.1
DA-TTA(ours)	**86.8**	**88.6**

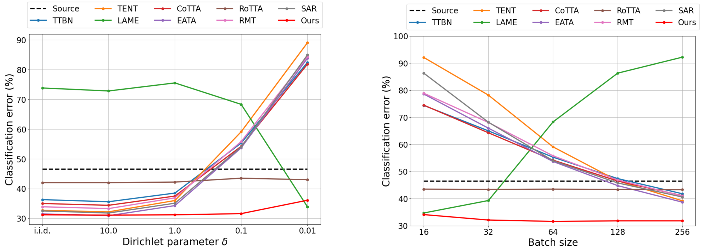
Table 4: Continual test-time adaptation in non-i.i.d. test data stream. Presented are the classification error rates (%) for TTA methods that are continually adapting to 15 target domains within the ImageNet-C dataset under a non-i.i.d. data stream scenario.

Method	——— **Continual adaptation to sequential domains** ——→															Mean
	bri	gla	jpe	con	def	imp	mot	sno	zoo	fro	pix	gau	ela	sho	fog	
Source	41.1	90.2	68.4	94.6	82.1	98.2	85.2	83.1	77.5	76.7	79.4	97.8	83.0	97.1	75.6	82.0
EATA [40]	95.1	99.5	99.5	99.8	99.8	99.8	99.8	99.7	99.8	99.8	99.8	99.9	99.8	99.8	99.8	99.4
TENT [55]	92.7	98.9	99.1	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.2
SAR [41]	92.0	98.7	97.8	99.6	99.7	99.7	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.0
CoTTA [57]	91.8	98.0	96.4	98.2	98.7	98.7	99.0	99.1	99.2	99.4	99.3	99.4	99.5	99.5	99.6	98.4
RMT [11]	93.6	98.2	96.5	98.2	98.5	98.2	98.5	98.3	98.5	98.7	97.7	98.5	97.7	98.7	97.6	97.8
TTBN [39, 48]	91.1	98.1	94.8	97.4	98.0	97.5	96.6	95.1	95.4	94.8	94.3	97.5	95.1	97.4	93.6	95.8
DELTA [73]	50.9	90.2	78.9	96.6	94.1	93.2	93.6	92.5	91.1	94.4	90.8	95.7	93.5	95.7	95.2	89.8
RoTTA [69]	37.7	86.6	65.0	96.1	85.7	89.9	88.4	84.7	86.2	87.0	84.0	94.1	87.4	93.0	96.6	84.2
LAME [3]	**27.6**	**83.3**	**45.6**	90.2	**57.3**	98.9	**71.6**	75.7	**60.1**	**52.6**	62.6	98.8	78.4	97.7	60.4	70.7
DA-TTA (ours)	36.9	88.9	59.4	**78.4**	78.3	**82.5**	74.1	**67.2**	63.8	73.4	**55.4**	**81.8**	**56.9**	**79.3**	**52.1**	**68.6**

Additionally, Tab. 3 shows the results on the ImageNet-A dataset. Our method demonstrates effectiveness in adapting to the adversarial attack domain shift.

Continual TTA in Non-I.I.D. Data Streams. In Tab. 4, we present the comparison of our method with other TTA methods when applied to a test data stream comprised of continual domains. This demanding stream delivers a sequence of 750 000 images from 15 target domains of ImageNet-C under the non-i.i.d. sampling condition. It is observed that the majority of competing methods present incur error rates in excess of 90%, significantly underperforming when compared to the "Source". While LAME performs optimally on a few target domains, it encounters failures in several domains, registering error rates above 90%. In comparison, DA-TTA showcases robust adaptation capabilities on all target domains and achieves the best overall average performance.

Robustness on Different Conditions of Data Streams. We examine the effect of the non-i.i.d. degree and the batch size. As illustrated in Fig. 5a, a smaller Dirichlet parameter δ indicates a higher degree of temporal correlation within the data stream. Most existing TTA methods experience a marked performance decline as δ decreases. LAME excels under intense temporal correlation, yet it underperforms compared to the baseline in less severe cases. In contrast, our method maintains robust performance across various degrees of non-i.i.d.



(a) Effect of Dirichlet parameter. **(b)** Effect of batch size.

Fig. 5: Robustness on various conditions of test data stream.

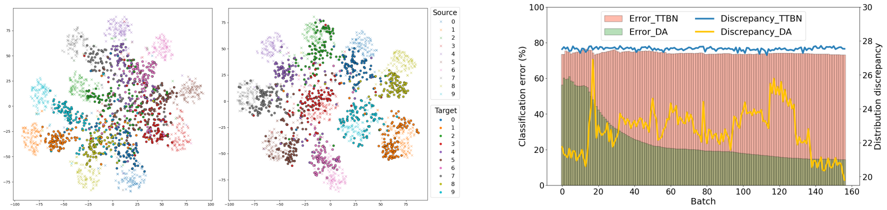
severity. Fig. 5b illustrates the impact of varying batch sizes. It's observed that most existing TTA methods experience a decline in performance as the batch size is reduced. This phenomenon could be attributed to larger batches more accurately representing the target domain's distribution, thereby reducing the conflict with the optimization objective. In contrast, our method shows consistent performance, proving to be robust across different batch sizes.

4.4 Ablation Studies

Effects of Model Components. As detailed in Tab. 5, we conduct an ablation study in non-i.i.d. data streams across three datasets. Firstly, we explored the effects of applying DA optimization within different ranges of the source model. The terms '*w/o* low-level DA' and '*w/o* high-level DA' refer to the application of DA optimization to the latter and former halves of the affine layers in the model, respectively. The results indicate a performance decline when the DA optimization range is reduced. However, the decrease in performance is relatively modest. This can be attributed to the correlated nature of feature distributions among layers within the frozen model, ensuring distributions in layers not directly supervised remain controlled. Moreover, the application of EM on top of the TTBN baseline, which is the TENT method, yields diminished results in non-i.i.d. data streams, as shown in Tab. 1. Nevertheless, introducing an EM loss atop the DA loss resulted in enhanced performance, highlighting the synergistic effect of the EM loss under protection from DA optimization.

Table 5: Ablation study in error rates. '*w/o* low-level DA' and '*w/o* high-level DA' denote removing the DA supervision from the first and second halves of the model.

Method	CIFAR10-C	CIFAR100-C	ImageNet-C
Source	43.5	46.5	82.0
w/o low-level DA	26.6	32.6	68.1
w/o high-level DA	24.9	33.5	67.1
w/o $\mathcal{L}_{EM}$	28.1	35.8	70.1
Ours	**24.3**	**31.6**	**64.8**



(a) **(b)**

Fig. 6: (a) DA alleviates domain shifts. Visualization of features prior to the classifier of the model for both source data and target data (one of target domains in CIFAR10-C dataset). Left: Source/target data inputs source model. Right: Source data inputs source model, and target data inputs source model with DA. (b) Verifying alignment between DA optimization and task objective from correlation between distribution discrepancy and classification error.

DA Alleviates Domain Shifts. In this analysis, we input source data and target domain data into the source model, labeling the features prior to the classifier layer as $\mathbf{R}_S$ for the source and $\mathbf{R}_T$ for the target, respectively. We then feed target data under our TTA method, labeling these features as $\mathbf{R'}_T$. Visualization of $\mathbf{R}_S$ and $\mathbf{R}_T$ is provided in the left plot in Fig. 6a via t-SNE, while the right plot depicts the $\mathbf{R}_S$ and $\mathbf{R'}_T$. Upon examining the transition from the left to the right, we observe a clear trend: the features of each class in the target domain not only become discriminative but also show an alignment with their corresponding classes in the source domain. This convergence of class-specific clusters confirms that our method is successfully reducing domain shift by steering the target feature distributions back to those from the source.

Alignment Between DA Optimization and Task Objective. DA optimization minimizes the discrepancy in distribution between test-time and source features. The task objective is to classify the incoming data stream. Fig. 6b provides a visualization of both the distribution discrepancy and the cumulative classification errors across data batches. Notably, there is a trend of decreasing accumulated error, which corresponds with the shrunken distribution discrepancy, in contrast to the larger discrepancy observed in the TTBN baseline.

5 Conclusion

In this paper, we introduce a simple yet effective Distribution Alignment (DA) method for fully realizing test-time adaptation within dynamic online streams. Our proposed distribution alignment loss aligns test-time data features with the source distributions, ensuring compatibility with the source model and addressing the challenges posed by label shifts across online data batches. The addition of a domain shift detection mechanism further strengthens our method's performance in environments with continual domain shifts. Extensive experiments confirm the superiority of our method in non-i.i.d. streams, while it also maintains competitive performance under the i.i.d. assumption.

Acknowledgements

This work is supported by an NSERC Discovery grant, the Gina Cody Research and Innovation Fellowship, and in part by the National Natural Science Foundation of China under Grant 62171269.

References

1. Ahmed, S., Al Arafat, A., Rizve, M.N., Hossain, R., Guo, Z., Rakin, A.S.: Ssda: Secure source-free domain adaptation. In: ICCV. pp. 19180–19190 (2023) 4
2. Bartler, A., Bühler, A., Wiewel, F., Döbler, M., Yang, B.: Mt3: Meta test-time training for self-supervised test-time adaption. In: International Conference on Artificial Intelligence and Statistics. pp. 3080–3090 (2022) 4
3. Boudiaf, M., Mueller, R., Ben Ayed, I., Bertinetto, L.: Parameter-free online test-time adaptation. In: CVPR. pp. 8344–8353 (2022) 4, 9, 11, 12, 24, 25
4. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: CVPR. pp. 3339–3348 (2018) 3
5. Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., Tang, J.: Metafscil: A meta-learning approach for few-shot class incremental learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2022) 9
6. Chi, Z., Gu, L., Zhong, T., Liu, H., YU, Y., Plataniotis, K.N., Wang, Y.: Adapting to distribution shift by visual domain prompt generation. In: International Conference on Learning Representations 2
7. Chi, Z., Mohammadi Nasiri, R., Liu, Z., Lu, J., Tang, J., Plataniotis, K.N.: All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In: European Confererence on Computer Vison (2020) 1
8. Chi, Z., Wang, Y., Yu, Y., Tang, J.: Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2021) 2
9. Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. In: NeurIPS (2021) 10
10. Ding, N., Xu, Y., Tang, Y., Xu, C., Wang, Y., Tao, D.: Source-free domain adaptation via distribution estimation. In: CVPR. pp. 7212–7222 (2022) 4
11. Döbler, M., Marsden, R.A., Yang, B.: Robust mean teacher for continual and gradual test-time adaptation. In: CVPR. pp. 7704–7714 (2023) 2, 3, 4, 5, 8, 9, 10, 11, 12, 22, 24, 25
12. Eastwood, C., Mason, I., Williams, C., Schölkopf, B.: Source-free adaptation to measurement shift via bottom-up feature restoration. In: ICLR (2022) 4, 10, 24
13. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV. pp. 2960–2967 (2013) 1
14. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. pp. 1180–1189 (2015) 4
15. Gao, J., Zhang, J., Liu, X., Darrell, T., Shelhamer, E., Wang, D.: Back to the source: Diffusion-driven adaptation to test-time corruption. In: CVPR. pp. 11786–11796 (2023) 4
16. Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., Lee, S.J.: Note: Robust continual test-time adaptation against temporal correlation. In: NeurIPS. vol. 35, pp. 27253–27266 (2022) 2, 3, 4, 5, 9, 10, 11, 23, 24

17. Hamidi, S.M., Tan, R., Ye, L., Yang, E.H.: Fed-it: Addressing class imbalance in federated learning through an information-theoretic lens. In: 2024 IEEE International Symposium on Information Theory (ISIT) (2024) 3
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1, 10
19. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., Gilmer, J.: The many faces of robustness: A critical analysis of out-of-distribution generalization. ICCV (2021) 10
20. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: ICLR (2019) 10
21. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. CVPR (2021) 10
22. Huang, J., Guan, D., Xiao, A., Lu, S.: Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In: NeurIPS. vol. 34, pp. 3635–3649 (2021) 4
23. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015) 8
24. Kang, G., Jiang, L., Yang, Y., Hauptmann, A.G.: Contrastive adaptation network for unsupervised domain adaptation. In: CVPR. pp. 4893–4902 (2019) 3
25. Koh, P.W., Sagawa, S., Marklund, H., Xie, S.M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R.L., Gao, I., et al.: Wilds: A benchmark of in-the-wild distribution shifts. In: ICML. pp. 5637–5664 (2021) 1
26. Kundu, J.N., Kulkarni, A.R., Bhambri, S., Mehta, D., Kulkarni, S.A., Jampani, V., Radhakrishnan, V.B.: Balancing discriminability and transferability for source-free domain adaptation. In: ICML. pp. 11710–11728 (2022) 3, 4
27. Kurmi, V.K., Subramanian, V.K., Namboodiri, V.P.: Domain impression: A source data free domain adaptation method. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 615–625 (2021) 3
28. Li, S., Xie, M., Gong, K., Liu, C.H., Wang, Y., Li, W.: Transferable semantic augmentation for domain adaptation. In: CVPR. pp. 11516–11525 (2021) 3
29. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: ICML. pp. 6028–6039 (2020) 2, 3, 4
30. Lim, H., Kim, B., Choo, J., Choi, S.: TTN: A domain-shift aware batch normalization in test-time adaptation. In: ICLR (2023) 4, 5
31. Lin, H., Zhang, Y., Qiu, Z., Niu, S., Gan, C., Liu, Y., Tan, M.: Prototype-guided continual adaptation for class-incremental unsupervised domain adaptation. In: ECCV. pp. 351–368 (2022) 3
32. Liu, H., Chi, Z., Yu, Y., Wang, Y., Chen, J., Tang, J.: Meta-auxiliary learning for future depth prediction in videos. In: IEEE Winter Conference on Applications of Computer Vision (2023) 2
33. Liu, H., Gu, L., Chi, Z., Wang, Y., Yu, Y., Chen, J., Tang, J.: Few-shot class-incremental learning via entropy-regularized data-free replay. In: European Conference on Computer Vision (2022) 9
34. Liu, Y., Kothari, P., Van Delft, B., Bellot-Gurlet, B., Mordan, T., Alahi, A.: Ttt++: When does self-supervised test-time training fail or thrive? In: NeurIPS. vol. 34, pp. 21808–21820 (2021) 4
35. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: ICML. pp. 97–105 (2015) 3

36. Long, M., Cao, Z., Wang, J., Jordan, M.I.: Conditional adversarial domain adaptation. In: NeurIPS. vol. 31 (2018) 4
37. Marsden, R.A., Döbler, M., Yang, B.: Introducing intermediate domains for effective self-training during test-time. arXiv preprint arXiv:2208.07736 (2022) 4, 5
38. Marsden, R.A., Döbler, M., Yang, B.: Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. arXiv preprint arXiv:2306.00650 (2023) 4, 5
39. Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., Snoek, J.: Evaluating prediction-time batch normalization for robustness under covariate shift. arXiv preprint arXiv:2006.10963 (2020) 2, 4, 5, 9, 11, 12, 24, 25
40. Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., Tan, M.: Efficient test-time model adaptation without forgetting. In: ICML. pp. 16888–16905 (2022) 2, 4, 5, 8, 9, 10, 11, 12, 24, 25
41. Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., Tan, M.: Towards stable test-time adaptation in dynamic wild world. In: ICLR (2023) 9, 10, 11, 12, 24, 25
42. Pei, Z., Cao, Z., Long, M., Wang, J.: Multi-adversarial domain adaptation. In: AAAI. vol. 32 (2018) 4
43. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV. pp. 1406–1415 (2019) 3
44. Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K., Wang, B.: Moment matching for multi-source domain adaptation. In: ICCV. pp. 1406–1415 (2019) 10
45. Purushotham, S., Carvalho, W., Nilanon, T., Liu, Y.: Variational recurrent adversarial deep domain adaptation. In: ICLR (2016) 4
46. Rusak, E., Schneider, S., Gehler, P.V., Bringmann, O., Brendel, W., Bethge, M.: Imagenet-d: A new challenging robustness dataset inspired by domain adaptation. In: ICML 2022 Shift Happens Workshop (2022) 10, 21
47. Sanyal, S., Asokan, A.R., Bhambri, S., Kulkarni, A., Kundu, J.N., Babu, R.V.: Domain-specificity inducing transformers for source-free domain adaptation. In: ICCV. pp. 18928–18937 (2023) 4
48. Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., Bethge, M.: Improving robustness against common corruptions by covariate shift adaptation. In: NeurIPS. vol. 33, pp. 11539–11551 (2020) 2, 4, 5, 9, 11, 12, 25
49. Shu, R., Bui, H.H., Narui, H., Ermon, S.: A dirt-t approach to unsupervised domain adaptation. In: ICLR (2018) 2
50. Song, J., Lee, J., Kweon, I.S., Choi, S.: Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In: CVPR. pp. 11920–11929 (2023) 2, 4, 5, 9, 10, 12, 24, 25
51. Storkey, A., et al.: When training and test sets are different: characterizing learning transfer. Dataset shift in machine learning **30**(3-28), 6 (2009) 5
52. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV. pp. 443–450 (2016) 3
53. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: ICML. pp. 9229–9248 (2020) 2, 4
54. Tang, L., Li, K., He, C., Zhang, Y., Li, X.: Consistency regularization for generalizable source-free domain adaptation. In: ICCV. pp. 4323–4333 (2023) 4
55. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: ICLR (2021) 1, 2, 4, 5, 9, 10, 11, 12, 25
56. Wang, F., Han, Z., Gong, Y., Yin, Y.: Exploring domain-invariant parameters for source free domain adaptation. In: CVPR. pp. 7151–7160 (2022) 4

57. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: CVPR. pp. 7201–7211 (2022) 2, 3, 4, 5, 9, 11, 12
58. Wilson, G., Cook, D.J.: A survey of unsupervised deep domain adaptation. ACM Transactions on Intelligent Systems and Technology **11**(5), 1–46 (2020) 3
59. Wu, Y., Chi, Z., Wang, Y., Feng, S.: Metagcd: Learning to continually learn in generalized category discovery. In: IEEE International Conference on Computer Vision (2023) 2
60. Wu, Y., Chi, Z., Wang, Y., Plataniotis, K.N., Feng, S.: Test-time domain adaptation by learning domain-aware batch normalization. In: AAAI Conference on Artificial Intelligence (2024) 2
61. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR (July 2017) 10
62. Yang, E.H., Hamidi, S.M., Ye, L., Tan, R., Yang, B.: Conditional mutual information constrained deep learning for classification. arXiv preprint arXiv:2309.09123 (2023) 1
63. Yang, E.H., Ye, L.: How to train the teacher model for effective knowledge distillation. In: European Conference on Computer Vision. Springer (2024) 3
64. Yang, E.H., Ye, L.: Markov knowledge distillation: Make nasty teachers trained by self-undermining knowledge distillation fully distillable. In: European Conference on Computer Vision. Springer (2024) 3
65. Yang, S., Jui, S., van de Weijer, J., et al.: Attracting and dispersing: A simple approach for source-free domain adaptation. In: NeurIPS. vol. 35, pp. 5802–5815 (2022) 4
66. Yang, S., Wang, Y., Van De Weijer, J., Herranz, L., Jui, S.: Generalized source-free domain adaptation. In: ICCV. pp. 8978–8987 (2021) 4
67. Ye, L., Hamidi, S.M., Tan, R., YANG, E.H.: Bayes conditional distribution estimation for knowledge distillation based on conditional mutual information. In: The Twelfth International Conference on Learning Representations (2024) 1
68. Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., Finn, C.: Gradient surgery for multi-task learning. In: NeurIPS. vol. 33, pp. 5824–5836 (2020) 7
69. Yuan, L., Xie, B., Li, S.: Robust test-time adaptation in dynamic scenarios. In: CVPR. pp. 15922–15932 (2023) 2, 3, 4, 5, 9, 10, 11, 12, 24, 25
70. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: Wilson, R.C., Hancock, E.R., Smith, W.A.P. (eds.) BMVC (2016) 10
71. Zhang, M., Levine, S., Finn, C.: Memo: Test time robustness via adaptation and augmentation. In: NeurIPS. vol. 35, pp. 38629–38642 (2022) 2, 4, 9, 11, 22, 24
72. Zhang, Y., Kang, B., Hooi, B., Yan, S., Feng, J.: Deep long-tailed learning: A survey. IEEE TPAMI **45**(9), 10795–10816 (2023) 3
73. Zhao, B., Chen, C., Xia, S.T.: Delta: Degradation-free fully test-time adaptation. In: ICLR (2023) 2, 4, 9, 10, 11, 12, 23
74. Zhong, T., Chi, Z., Gu, L., Wang, Y., Yu, Y., Tang, J.: Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. Advances in Neural Information Processing Systems (2022) 2
75. Zhou, Z., Guo, L.Z., Jia, L.H., Zhang, D., Li, Y.F.: Ods: Test-time adaptation in the presence of open-world data shift. In: ICML (2023) 4