ODS: Test-Time Adaptation in the Presence of Open-World Data Shift

Zhi Zhou [1] **Lan-Zhe Guo** [1] **Lin-Han Jia** [1] **Ding-Chu Zhang** [1] **Yu-Feng Li** [1]

Abstract

Test-time adaptation (TTA) adapts a source model to the distribution shift in testing data without using any source data. There have been plenty of algorithms concentrated on covariate shift in the last decade, i.e., $\mathcal{D}_t(X)$, the distribution of the test data is different from the source data. Nonetheless, in real application scenarios, it is necessary to consider the influence of label distribution shift, i.e., both $\mathcal{D}_t(X)$ and $\mathcal{D}_t(Y)$ are shifted, which has not been sufficiently explored yet. To remedy this, we study a new problem setup, namely, TTA with Open-world Data Shift (AODS). The goal of AODS is simultaneously adapting a model to covariate and label distribution shifts in the test phase. In this paper, we first analyze the relationship between classification error and distribution shifts. Motivated by this, we hence propose a new framework, namely ODS, which decouples the mixed distribution shift and then addresses covariate and label distribution shifts accordingly. We conduct experiments on multiple benchmarks with different types of shifts, and the results demonstrate the superior performance of our method against the state of the arts. Moreover, ODS is suitable for many TTA algorithms.

1. Introduction

Deep neural networks (DNNs) have achieved great success in many application scenarios, such as computer vision (He et al., 2016; Krizhevsky et al., 2012), speech recognition (Amodei et al., 2016), and natural language processing (Chowdhury, 2003). These successes typically rely on the independent identically distribution (i.i.d.) assumption that training and testing data are drawn from the same distribution. In practice, this assumption is difficult to hold,

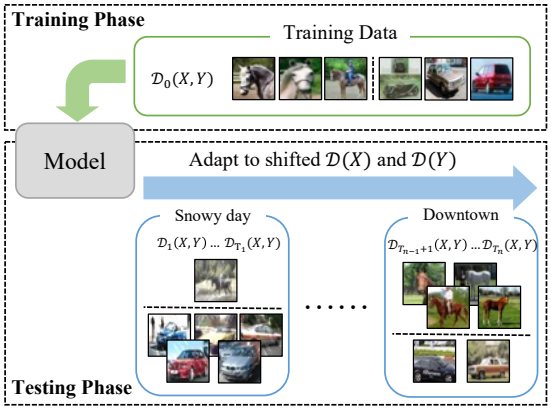


Figure 1. A demonstration of the AODS setup. The pre-trained model will continually adapt to the changing covariate and label distributions in the testing phase.

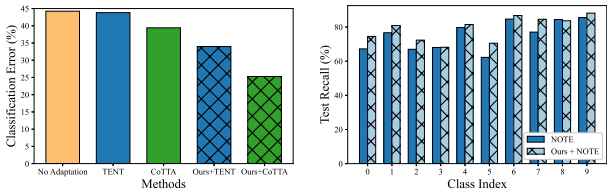
e.g., change of weather, scenes and sensor devices will cause distribution shifts (Hendrycks & Dietterich, 2019). Therefore, the distribution shift has a severe impact on the performance of Deep Neural Networks (DNNs), which has become a critical issue.

When dealing with distribution shifts in the testing phase, plenty of test-time adaptation (TTA) algorithms have been proposed to improve the performance of DNNs (Nado et al., 2020; Liu et al., 2021; Zhang et al., 2021; Shin et al., 2022; Kim et al., 2022). Existing algorithms mainly work by adapting the model parameters (Wang et al., 2021; Niu et al., 2022) or optimizing the predictions (Wang et al., 2022a; Boudiaf et al., 2022) with only unlabeled testing data. Note that these methods are particularly designed to deal with covariate shifts but ignore that label distribution $\mathcal{D}_t(Y)$ also accordingly shifts. Some other works, such as the robust TTA method (Gong et al., 2022), also consider the variation of $\mathcal{D}_t(Y)$ during the test period. However, they only try to reduce the adverse effect caused by $\mathcal{D}_t(Y)$, but do not adapt to it.

Although existing TTA methods significantly improve performance, they still ignore the adaptation to the changing label distribution during the testing phase, which is crucial for practical machine learning systems (des Combes et al., 2020; Zhao et al., 2021; Wu et al., 2021; Bai et al., 2022).

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: Yu-Feng Li <liyf@nju.edu.cn>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).



(a) Classification Error (%) (b) Test Recall(%)

Figure 2. Example of experimental results on the CIFAR10 dataset with label distribution shifts during testing. (a) Changing the label distribution makes existing TTA methods almost useless. The proposed framework can efficiently adapt to the changed label distribution, thereby boosting existing TTA methods. (b) The robust TTA method cannot adapt to the changed label distribution. With our framework, the recalls of most classes are improved.

For example, a flu prediction model trained on data with a regular morbidity rate will not perform well in a location or over a period with a high morbidity rate due to label distribution shift (Tasche, 2017). Computer vision applications, such as predicting object locations (Yang et al., 2018) and human poses (Martinez et al., 2017), also experience changes in label distribution. In this paper, we investigate the practical problem of simultaneously tuning the model for covariance and label distribution changes during testing. We name this TTA with Open-world Data Shift (AODS). An illustration of the AODS can be found in Figure 1. Figure 2(a) shows that the TTA method does not gain performance when the covariate and the label distribution are changed together. Figure 2(b) further shows that there is a large room for performance improvement if the TTA method adapts to the shifted label distribution. Therefore, AODS is of great importance and remains challenging.

To address the AODS problem, we first analyze the relationship between classification error and distribution shifts. We discover that adapting the feature representation and optimizing the model prediction according to estimated label distribution help reduce the classification error. Motivated by this theoretical analysis, we hence present Open-world Data Shift adaptation (ODS), a generic framework for simultaneously adapting a model to changing covariate and label distributions in the testing phase. Different from previous TTA methods that either neglect the changing label distribution (Wang et al., 2021; Niu et al., 2022) or forcibly balance it (Gong et al., 2022), ODS tracks label distribution and then take full advantage of it. Specifically, ODS contains two essential modules: *Distribution Tracker* $\mathcal{M}_T$ for estimating the label distribution $\boldsymbol{w}_t$ at each timestamp t and *Prediction Optimizer* $\mathcal{M}_O$ for optimizing the model predictions based on the estimated label distribution. On one hand, we handle the adverse effects caused by shifted $\mathcal{D}_t(Y)$ with the help of $\boldsymbol{w}_t$ in an adaptive adaptation formulation. On the other hand, we optimize the model prediction to be optimal, making it consistent with the corresponding distribution. Furthermore, the proposed ODS framework is applicable to many TTA methods. The integrated TTA methods and two essential modules jointly optimize to improve test-time performance.

The contributions of this paper are as follows:

1) We investigate a novel problem of tuning the model to accommodate both covariance and label changes during the testing, which is practical in real-world applications.

2) Based on our theoretical analysis, we propose a new TTA framework. It efficiently adapts the model to open-world data shift through two basic modules and can be easily integrated with many existing TTA algorithms.

3) We evaluate the algorithm on multiple benchmarks with varying degrees of shifts, which shows that the proposed scheme significantly outperforms state-of-the-art TTA methods.

2. Problem and Analysis

In this section, we first describe the notations and problem formulation of AODS. Then, a theoretical analysis presents the relationship between classification error and distribution shifts, which strongly motivates our algorithm design.

2.1. Problem Formulation

We focus on the multi-class classification with input space $\mathcal{X} \in \mathbb{R}^{\boldsymbol{d}}$ and label space $\mathcal{Y} = [K] \triangleq \{1, \dots, K\}$, where $\boldsymbol{d}$ and K represent input dimensions and number of classes, respectively. Accordingly, we use X, Y, Z to denote random variables of samples, labels, and feature representations. $\mathcal{D}_t(X)$, $\mathcal{D}_t(Y)$, and $\mathcal{D}_t(X, Y)$ indicate covariate distribution, label distribution, and joint distribution at each timestamp t, respectively.

In AODS problem, we are given a source model $f_{\theta_0} : \mathcal{X} \mapsto \mathcal{Y}$ well trained on class-balanced source data $\mathcal{D}_0(X, Y)$ with initial parameters θ_0. The probability of prediction is denoted as $f_{\theta_0}(Y|X) : \mathcal{X} \mapsto \mathbb{R}^K$. We deploy the model into the actual applications, where covariate distribution $\mathcal{D}_t(X)$ and label distribution $\mathcal{D}_t(Y)$ constantly change. At each timestamp t, the model gives the predictions and then continually evolves its parameter $\theta_t \rightarrow \theta_{t+1}$ using unlabeled testing data. The goal of AODS is to simultaneously adapt the model to the time-varying covariate distribution $\mathcal{D}_t(X)$ and label distribution $\mathcal{D}_t(Y)$, namely open-world data shift, for better performance in the testing phase.

Without prior knowledge or assumptions, it is impossible to adapt the source model to changing distributions. Following previous studies (des Combes et al., 2020; Wang

et al., 2022a), we consider the data distribution periodically changes under generalized label shift in AODS problem:

Definition 2.1 (Generalized Label Shift, GLS)**.** Both covariate distribution $\mathcal{D}_0(X) \neq \mathcal{D}_t(X)$ and label distribution $\mathcal{D}_0(Y) \neq \mathcal{D}_t(Y)$ change. Meanwhile, there exists a feature representation $Z = g^\star(X)$ satisfies

$$\mathcal{D}_0(Z|Y=y) = \mathcal{D}_t(Z|Y=y), \forall y \in \mathcal{Y} \tag{1}$$

Remark 2.2. Although Definition 2.1 allows $\mathcal{D}_t(X)$ and $\mathcal{D}(Y)$ to differ from the source data during testing, the existence of an intermediate feature representation $g^\star(X)$ ensures that the TTA algorithm can maintain good performance. Taking the example in Figure 1, samples from the same class may differ dramatically at different moments. If the TTA algorithm adapts the representation extractor to $g^\star$, then it can fit the covariate distribution and track the label distribution at each timestamp.

2.2. Problem Analysis

Empirical results in Figure 2 and Theorem 3.1 in (des Combes et al., 2020) show that learning methods that do not adapt to changing label distributions will increase their classification error. To analyze the AODS problem theoretically, we assume that the label distribution $\mathcal{D}_t(Y)$ at each moment can be estimated by $\boldsymbol{w}_t$. Then, we optimize the predictions by the plug-in rule introduced in (Menon et al., 2013) to account for the changing $\mathcal{D}_t(Y)$. To simplify the notation, we denote the original model prediction as $\hat{Y} = f_{\theta_t}(X)$ on X for any model f_{θ_t}. The optimized prediction introduced above is expressed as:

$$\hat{Y}_o = \arg\max_{y \in \mathcal{Y}} f_{\theta_t}(Y=y|X) + \ln w_{t,k} \tag{2}$$

This plug-in rule effectively corrects mismatched label distributions (Menon et al., 2021) and has good statistical properties (Menon et al., 2013; Collell et al., 2016).

We then attempt to understand the effectiveness of the above optimizations through theoretical analysis. Performance guarantees are provided to limit the error gap between the source data distribution $\mathcal{D}_0(X,Y)$ and the data distribution $\mathcal{D}_t(X,Y)$ at each timestamp t of the adapted model. We first explain some terms as follows. Given a model f_{θ_t} evaluated on data distribution $\mathcal{D}_t(X,Y)$, its error is $\varepsilon_t(\hat{Y}) = \mathcal{D}_t(\hat{Y} \neq Y)$ and its error of optimized prediction is $\varepsilon_t(\hat{Y}_o) = \mathcal{D}_t(\hat{Y}_o \neq Y)$. We define the balanced source error as $BSE\left(\hat{Y}\right) = \max_{y \in \mathcal{Y}} \mathcal{D}_0\left(\hat{Y} \neq y | Y = y\right)$ and conditional error gap between $\mathcal{D}_0(X,Y)$ and $\mathcal{D}_t(X,Y)$ as

$$\Delta_{CE}(\hat{Y}) = \max_{y \neq y' \in \mathcal{Y}} |\mathcal{D}_0(\hat{Y}=y'|Y=y) - \mathcal{D}_t(\hat{Y}=y'|Y=y)| \tag{3}$$

$BSE(\hat{Y})$ measures the model performance of f_{θ_t} on source data $\mathcal{D}_0(X,Y)$. $\Delta_{CE}(\hat{Y})$ measures the generalization of feature representations adapted by TTA algorithm. When TTA algorithm adapts their feature representation exactor to $g^\star$ in Definition 2.1, $\Delta_{CE}(\hat{Y})$ is equal to 0.

Based on the above terms, we give an upper bound on the error gap between $\mathcal{D}_0(X,Y)$ and $\mathcal{D}_t(X,Y)$:

Theorem 2.3. *For any model f_{θ_t}, the error gap $\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right|$ is upper bounded by*

$$C\|\mathbf{1} - \frac{\mathcal{D}_t(Y)}{\boldsymbol{w}_t}\|_1 BSE\left(\hat{Y}\right) + 2(K-1)\Delta_{CE}(\hat{Y}) \tag{4}$$

where C is a constant related to $\mathcal{D}_0(Y)$.

Remark 2.4. Theorem 2.3 decomposes the error gap of f_{θ_t} between $\mathcal{D}_0(X,Y)$ and $\mathcal{D}_t(X,Y)$. This decomposition is more informative than a direct comparison to the optimal model, since the optimal performance is unknown and changes gradually as the distribution changes. Compared to results in previous studies (des Combes et al., 2020), our results additionally deal with time-varying label distributions. When we do not track the label distribution and set $\boldsymbol{w}_t$ to a uniform distribution, our theoretical results are consistent with previous studies.

Remark 2.5. The first term in the upper bound contains $\|\mathbf{1} - \mathcal{D}_t(Y)/\boldsymbol{w}_t\|_1$, which measures the distance between the ground-truth label distribution $\mathcal{D}_t(Y)$ and its estimate $\boldsymbol{w}_t$, indicating that more accurate estimates $\boldsymbol{w}_t$ lead to better performance. The first term also contains $BSE\left(\hat{Y}\right)$, which shows the relationship between the performance of the current distribution and the performance of the source data $\mathcal{D}_0(X,Y)$. This also explains why TTA methods (Niu et al., 2022; Wang et al., 2022a), which try to prevent catastrophic forgetting, perform better. The second term in the upper bound $\Delta_{CE}(\hat{Y})$ measures the effectiveness of the feature representation adapted by the TTA algorithm.

In summary, Theorem 2.3 motivates us to keep track of changing label distributions, while adaptively performing model adaptation and efficiently optimizing predictions so that models can adapt to both covariance and label distributions. The proof is shown in Appendix A.

3. Methodology

We propose the framework in this work: ODS, a general framework that enables TTA methods to simultaneously adapt to changing covariate and label distributions. The dilemma in achieving this goal is that, on one hand, directly fitting the model to the two distributions leads to severe performance degradation; on the other hand, removing the adverse effect of shifted label distribution instead of adapting it makes the predictions suboptimal.

To solve the above problems, we first propose two basic modules of ODS motivated by Theorem 2.3:

1) *Distribution Tracker* $\mathcal{M}_T$: Given unlabeled data at timestamp t, $\mathcal{M}_t$ estimates label distribution $\boldsymbol{w}_t$ for subsequent adaptation and predictive optimization.

2) *Prediction Optimizer* $\mathcal{M}_O$: Given a previously estimated $\boldsymbol{w}_t$ and an adjusted model f_{θ_t}, $\mathcal{M}_O$ improves the prediction that is distributionally consistent with the ground-truth label distribution $\mathcal{D}_t(Y)$.

We can then handle data shifts for covariate and label distributions separately using the following formulas:

$$\min_{\theta_t} \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{k=1}^{K} S(\boldsymbol{w}_t)_k f_{\theta_t}(Y=k|\boldsymbol{x}_i) \log f_{\theta_t}(Y=k|\boldsymbol{x}_i)$$
$$\text{s.t.} \quad \boldsymbol{w}_t \text{ is estimated by } \mathcal{M}_T \tag{5}$$

where N_t is the number of test samples at this timestamp and $S(\boldsymbol{w}_t) = \text{Normalize}(\mathbf{1} - \boldsymbol{w}_t)$ inversely weights each class in loss according to $\boldsymbol{w}_t$. The objective in (5) means that we explicitly track the changing label distribution and use it to help our test-time adaptation.

Finally, the prediction $\hat{Y}_o$ of our framework is improved by the module $\mathcal{M}_O$ and $\boldsymbol{w}_t$ estimated by $\mathcal{M}_T$:

$$\hat{Y}_o = \mathcal{M}_O\left(f_{\theta_t}(Y=k|\boldsymbol{x}), \boldsymbol{w}_t\right) \tag{6}$$

Equation (6) means that our framework maintains an internal model f_{θ_t} that gives balanced predictions and can be improved immediately with the help of $\mathcal{M}_O$ and the estimated label distribution $\boldsymbol{w}_t$. Eq.(2) is one implementation.

Our framework is plug-and-play with existing TTA algorithms. A general schematic description of the framework is shown in Figure 3. Below we describe in detail two of these basic modules: $\mathcal{M}_T$ and $\mathcal{M}_O$.

3.1. Distribution Tracker $\mathcal{M}_T$

Following the objective in (5), the first task is to estimate the label distribution at each timestamp t. For this purpose, Black Box Shift Estimation (BBSE) (Lipton et al., 2018) is a powerful technique. It uses the source model f_{θ_0} and its statistics (i.e., the estimated confusion matrix $\hat{\boldsymbol{C}}_{\hat{Y},Y}$) to estimate the label shift $\frac{\mathcal{D}_t(Y)}{\mathcal{D}_0(Y)}$. Let $\boldsymbol{\gamma}_t$ denote the label distribution estimated at timestamp t, which is evaluated by f_{θ_0} where $\boldsymbol{\gamma}_t = \frac{1}{N_t}\sum_{i=1}^{N_t} f_{\theta_0}(Y|\boldsymbol{x}_i)$. Then, BBSE generates label shift at timestamp t as follows:

$$\frac{\mathcal{D}_t(Y)}{\mathcal{D}_0(Y)} = \hat{\boldsymbol{C}}_{\hat{Y},Y}^{-1} \boldsymbol{\gamma}_t \tag{7}$$

In our TTA setting, we assume that the source model f_{θ_0} is well-trained on balanced source data. Therefore, the label distribution $\boldsymbol{w}_t$ estimated at timestamp t approximates $\boldsymbol{\gamma}_t$.

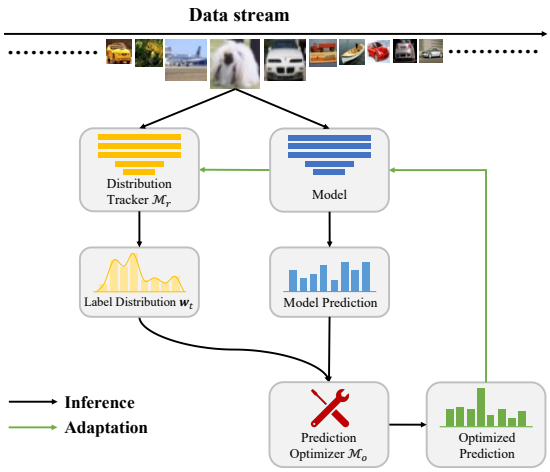


Figure 3. The proposed framework consists of two important parts that estimate and exploit the label distribution at each timestamp.

However, $\boldsymbol{\gamma}_t$ is a poor approximation of the ground-truth label distribution $\mathcal{D}_t(Y)$ because the covariate distribution changes, breaking the assumption $\mathcal{D}_0(X|Y) = \mathcal{D}_t(X|Y)$. Note that Definition 2.1 ensures that $\mathcal{D}_0(Z|Y) = \mathcal{D}_t(Z|Y)$, which inspires us to use the feature representation of f_{θ_t} to improve the estimation effect. Specifically, we use the semi-supervised learning technique (Zhou, 2018; Wang et al., 2022b) to optimize the label vector $\boldsymbol{z}_i$ of each sample $\boldsymbol{x}_i$, and then estimates the label distribution through $\boldsymbol{w}_t = \frac{1}{N_t}\sum_{i=1}^{N_t} \boldsymbol{z}_i$. Guided by the clustering assumption (Grandvalet & Bengio, 2004) and the smoothing assumption (Wagner et al., 2018), the objective optimizes $\boldsymbol{z}_i$ in an unsupervised manner using entropy loss and consistency loss. The optimization objective is formalized as:

$$\min_{\boldsymbol{w}_t} \sum_{i=1}^{N_t} \left[\boldsymbol{z}_i^\top \log f_{\theta_0}(Y|\boldsymbol{x}_i) + \boldsymbol{z}_i^\top \log \boldsymbol{z}_i - \sum_{j=1}^{N_t} s_{ij} \boldsymbol{z}_i^\top \boldsymbol{z}_j \right]$$
$$\text{s.t.} \quad \boldsymbol{w}_t = \frac{1}{N_t}\sum_{i=1}^{N_t} \boldsymbol{z}_i \tag{8}$$

satisfying constraints $\mathbf{1}^\top \boldsymbol{z}_i = 1, \forall i \in \{1, \ldots, N_t\}$. $s_{i,j}$ is the similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ measured on feature representations of newly adapted f_{θ_t}. We adopt the iterative solution yielded by (Boudiaf et al., 2022; Yuille & Rangarajan, 2001) for efficiently solving objective in Equation (8):

$$z_{i,k}^{(n+1)} = \frac{f_{\theta_0}(Y|\boldsymbol{x}_i) \exp\left(\sum_j s_{i,j} z_{j,k}^{(n)}\right)}{\sum_{k'} f_{\theta_0}(Y|\boldsymbol{x}_i) \exp\left(\sum_j s_{ij} z_{j,k'}^{(n)}\right)} \tag{9}$$

LAME (Boudiaf et al., 2022) uses a similar idea to optimize predictions. The difference is two folds: a) We are

motivated to improve the estimation of label distribution $\boldsymbol{w}_t$ obtained by the BBSE method instead of prediction; b) Definition 2.1 theoretically inspires us to adopt the feature representation of the newly adapted model for optimization. We also demonstrate empirically that the direct combination of the TTA method with LAME does not achieve satisfactory performance improvements in Section 4.4.

Furthermore, the estimates are not stable within each batch. Therefore, we use a fixed-length FIFO queue to cache the labels of recent samples and compute the cached average label distribution $\boldsymbol{w}_t$. This queue only needs to store labels and therefore consumes negligible resources. In our experiments, the queue length is set to $20\times$ the number of classes. The overall algorithm of $\mathcal{M}_T$ is shown in Algorithm 1.

3.2. Prediction Optimizer $\mathcal{M}_O$

We introduce the detail of $\mathcal{M}_O$, which uses the estimated label distribution $\boldsymbol{w}_t$ to make $f_{\theta_t}(Y|X)$ consistent with $\mathcal{D}_t(Y)$. One feasible approach is to directly adopt the plug-in solution in (2) to optimize the predictions introduced in Section 2.2. However, this simple strategy may lead to performance degradation in practice since it is not robust to errors in estimated $\boldsymbol{w}_t$, especially when the number of classes in the target task is large.

To address the above issues, we propose a conservative approach to optimize predictions $\hat{\boldsymbol{y}}_i$ for $\boldsymbol{x}_i$. We adopt ensemble strategy (Zhou, 2021) to take into account both the results of the original model prediction and estimated $\boldsymbol{w}_t$ in (8). Specifically, the original prediction $f_{\theta_t}(Y|\boldsymbol{x}_i)$ and the intermediate result $\boldsymbol{z}_i$ in (8), which represents estimated label distribution $\boldsymbol{w}_t$, are ensembled as follows:

$$\hat{\boldsymbol{y}}_{i,k} = \frac{\sqrt{\boldsymbol{z}_{i,k} f_{\theta_t}(Y=k|\boldsymbol{x}_i)}}{\sum_{k'\in\mathcal{Y}} \sqrt{\boldsymbol{z}_{i,k'} f_{\theta_t}(Y=k'|\boldsymbol{x}_i)}} \tag{10}$$

Then, the prediction is optimized as follows:

$$\hat{Y}_o = \arg\max_{k\in\mathcal{Y}} \hat{y}_k \tag{11}$$

This optimization process is efficient, and computation is almost negligible. We discuss the performance of two strategies in (2) and (11) in Section 4.4, as well as their time consumption.

4. Experiments

In this section, we conduct experiments to answer the following three research questions:

RQ1: Whether ODS can outperform prior TTA methods when encountering open-world data shift?

RQ2: Whether ODS is generic to integrate with different TTA methods and boost their performance?

Algorithm 1 Distribution Tracker $\mathcal{M}_T$

Input: source model f_{θ_0}, adapted model f_{θ_t}, samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_t}$
Output: label distribution $\boldsymbol{w}_t$
$que \leftarrow$ Global maintained FIFO queue with fixed size
for $i = 1$ **to** N_t **do**
  $\boldsymbol{p}_i \leftarrow f_{\theta_0}(Y=y|\boldsymbol{x}_i)$
  $\boldsymbol{F}_i \leftarrow$ Feature representations of $f_{\theta_t}(Y=y|\boldsymbol{x}_i)$
end for
Calculate similarity $s_{i,j}$ with $\{F_1, \ldots, F_{N_t}\}$
Calculate $\boldsymbol{Z} = [\boldsymbol{z}_1; \ldots; \boldsymbol{z}_{N_t}]$ according to Equation (9)
for $i = 1$ **to** N_t **do**
  Push $\arg\max_{k=\{1,\ldots,K\}} \boldsymbol{z}_{i,k}$ into que
end for
$\boldsymbol{w}_t \leftarrow$ Average label distribution of que
return $\boldsymbol{w}_t$

RQ3: Does ODS accurately estimate label distribution and effectively optimize the prediction?

4.1. Experimental Setup

Datasets. We conduct experiments on two standard TTA benchmarks: CIFAR10-C and CIFAR100-C (Hendrycks & Dietterich, 2019). For experiments on the CIFAR dataset, We train the source model on the clean CIFAR10/CIFAR100 dataset, which has 50,000 32×32 training images associated with 10/100 classes. Then, we test each TTA method on the CIFAR10-C/CIFAR100-C dataset, which contains 15 corrupt testing sets belonging to four categories. Similar to previous studies (Gong et al., 2022; Wang et al., 2021; 2022a), we report the results evaluated on the most severe corruption level of 5. To construct the shifted label distribution in benchmark datasets, we employ the tweak-one shift introduced in the previous study (Guo et al., 2020). For each corruption type, we will set the probability of one or several selected classes to appear γ times that of other classes to simulate the rising probability of certain classes in different scenarios. We conduct experiments with three distribution shift settings, i.e., $\gamma = 2$, $\gamma = 5$, and $\gamma = 10$. The detailed experimental setup is presented in Appendix B.

Compared Methods. We compare our ODS with the various TTA algorithms, including typical TTA methods, continual TTA methods, and recently proposed robust TTA methods. Specifically, for typical TTA methods, we take a wide range of studies into comparison: Test-time normalization (Schneider et al., 2020) (BN STATS) updates the statistics of BN layers from the batch of test samples; Test entropy minimization (Wang et al., 2021) (TENT) further updates the parameters of BN layers with entropy minimization loss; Efficient anti-forgetting test-time adap-

Table 1. Comparison with state-of-the-art TTA methods on CIFAR10 dataset with severity level 5 and $\gamma = 10$. We omit std in this table due to space issues. The bold number indicates the best results. ODS outperforms comparison methods on almost all corruptions.

METHODS	NOISE			BLUR				WEATHER				DIGITAL				AVG.
	GAUSS.	SHOT	IMPUL.	DEFOC.	GLASS	MOTION	ZOOM	SNOW	FROST	FOG	BRIT.	CONTR.	ELASTIC	PIXEL	JPEG	
SOURCE	14.70	18.52	15.61	56.92	31.99	68.01	63.25	82.19	72.44	76.31	**92.41**	23.38	72.33	68.72	79.72	55.77
BN STATS	50.60	51.16	45.31	71.73	47.99	69.35	68.59	60.16	60.39	64.27	69.60	67.56	59.21	66.12	58.17	60.68
TENT	53.53	60.97	59.34	63.33	47.12	65.81	68.11	55.08	55.00	58.68	63.40	49.59	46.95	50.45	45.38	56.18
EATA	48.94	48.21	42.05	65.44	43.42	59.81	57.27	55.09	52.98	56.00	59.54	61.47	51.32	55.75	50.88	53.88
LAME	57.99	60.15	53.07	78.83	53.04	76.67	74.90	67.81	67.30	71.94	77.05	74.84	68.53	73.44	66.90	68.16
COTTA	57.43	60.06	56.03	66.66	52.25	66.54	66.65	58.32	58.92	60.09	64.69	55.05	59.37	64.74	61.92	60.58
NOTE	51.90	54.57	68.38	84.29	50.53	88.97	86.21	86.15	86.68	83.27	86.48	90.64	77.84	80.77	81.02	77.18
ODS	**67.45**	**65.78**	**71.88**	**88.66**	**56.32**	**90.48**	**88.09**	**86.16**	**86.93**	**83.96**	87.37	**91.16**	**79.35**	**84.43**	**82.02**	**80.67**

tation (Niu et al., 2022) (EATA) performs active sample selection for adaptation to simultaneously achieve both high adaptation efficiency and strong predicting performance. For continual TTA methods, we compare one typical approach: Continual test-time adaptation (Wang et al., 2022a) (COTTA) eliminates the prediction error accumulated in the data stream via weight-and-augmentation-averaged pseudo-labels and parameters stochastic restoration. For robust TTA methods, we take two recently proposed SOTA methods into comparison: Laplacian adjusted maximum-likelihood estimation (Boudiaf et al., 2022) (LAME) adopts a conservative adaptation approach that modifies the model's outputs rather than the model's parameters during testing; Non-i.i.d. test-time adaptation scheme (Gong et al., 2022) (NOTE) proposes instance-aware batch normalization and prediction-balanced reservoir sampling to alleviate the unexpected effects of non-i.i.d. data streams.

Implementation Details. For all experiments, we adopt the ResNet18 (He et al., 2016) as the backbone since it is commonly used in previous studies (Boudiaf et al., 2022; Gong et al., 2022). We train the source model with batch size 256 for 200 epochs. The SGD optimizer optimizes each model with a learning rate of 0.1 using a cosine annealing schedule. For test-time adaptation, we set the batch size to 64 following previous studies (Wang et al., 2022a; Niu et al., 2022). For all comparison methods, we use their original hyperparameter in their paper. We report mean $\pm$ std accuracy over five runs with random seed setting to 0, 1, 2, 3, 4. This work uses the Huawei MindSpore platform for experimental testing partially. The other implementation details are presented in Appendix B.

4.2. Empirical Results

RQ1: Whether ODS can outperform prior TTA methods when encountering open-world data shift?

Our framework can integrate with many TTA methods. To demonstrate the effectiveness of ODS, we combine our proposal with the SOTA robust TTA method NOTE (Gong et al., 2022) to answer this question. Table 1 gives the detailed results on CIFAR-10 dataset with shift level $\gamma = 10$. We continually evaluate each method on all corruptions in order. The results show that ODS consistently outperforms existing TTA methods on almost every corruption and achieves 3.71% accuracy improvement compared to the SOTA method NOTE. We also evaluate ODS and comparison methods on CIFAR-10 and CIFAR-100 datasets with three shift levels. Table 2 and Table 3 show that ODS gives the best and most stable performance no matter the level of label shift because it actively tracks the label distribution and optimizes the prediction. While the other methods suffer from performance degradation when the shift level of label distribution changes. Although COTTA performs better than ODS on the CIFAR10 dataset with $\gamma = 2$, ODS still gives competitive results.

Table 2. Comparison with state-of-the-art TTA methods on CIFAR10 dataset with three shift levels. Bold indicates the best.

METHODS	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$
SOURCE	56.41 ± 0.05	56.12 ± 0.07	55.77 ± 0.16
BN STATS	78.33 ± 0.05	71.75 ± 0.08	60.68 ± 0.14
TENT	68.85 ± 3.14	66.94 ± 3.52	56.18 ± 4.13
EATA	79.35 ± 0.16	69.23 ± 0.25	53.88 ± 0.53
LAME	78.96 ± 0.05	75.20 ± 0.10	68.16 ± 0.13
COTTA	$\mathbf{81.81 \pm 0.37}$	73.58 ± 0.28	60.58 ± 0.15
NOTE	78.81 ± 0.27	77.96 ± 0.75	77.18 ± 0.38
ODS	81.13 ± 0.09	$\mathbf{80.40 \pm 0.36}$	$\mathbf{80.67 \pm 0.29}$

RQ2: Whether ODS is generic to integrate with different TTA methods and boost their performance?

To validate the universality of our framework, we apply ODS to three representative TTA methods: 1) TENT, a typical TTA method; 2) COTTA, a continual TTA method; 3) NOTE, current SOTA robust TTA method. The detailed results of each corruption are shown in Figure 4. ODS framework can consistently boost the performance of three TTA methods when we continually evaluate them on different corruptions. Table 4 presents the average results on different shift levels, demonstrating the effectiveness of the proposed ODS framework.

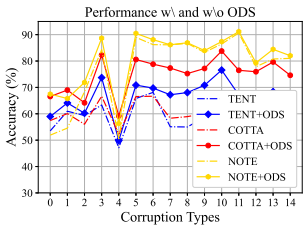


Figure 4. ODS framework effectively boosts TTA methods: TENT, COTTA, NOTE.

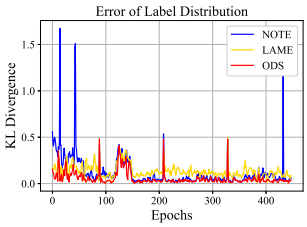


Figure 5. Comparison of estimated label distribution of NOTE, LAME, ODS.

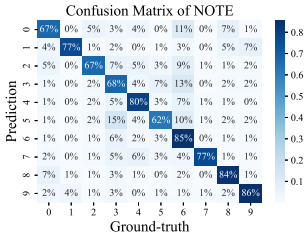


Figure 6. The confusion matrix of NOTE on CIFAR10 dataset.

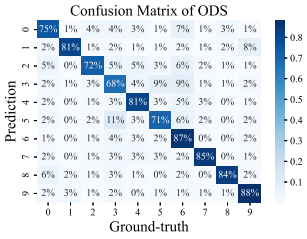


Figure 7. The confusion matrix of ODS on CIFAR10 dataset.

Table 3. Comparison with state-of-the-art TTA methods on CIFAR100 dataset with three shift levels. Bold indicates the best.

METHODS	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$
SOURCE	32.71 ± 0.15	32.71 ± 0.18	32.75 ± 0.14
BN STATS	52.69 ± 0.20	52.82 ± 0.08	52.76 ± 0.15
TENT	40.07 ± 2.35	51.39 ± 0.59	52.95 ± 0.17
EATA	43.68 ± 18.16	45.12 ± 15.79	48.99 ± 7.79
LAME	52.49 ± 0.25	52.51 ± 0.24	52.62 ± 0.21
COTTA	47.74 ± 0.59	50.48 ± 0.57	51.72 ± 0.47
NOTE	50.34 ± 0.11	48.41 ± 0.33	47.06 ± 0.35
ODS	$\mathbf{56.86 \pm 0.18}$	$\mathbf{56.43 \pm 0.21}$	$\mathbf{55.83 \pm 0.23}$

Table 4. Average performance of existing TTA methods w/ and w/o ODS framework. The bold number indicates the best result. ODS can consistently improve the performance of TTA methods.

METHODS	$\gamma = 2$	$\gamma = 5$	$\gamma = 10$
TENT	68.85 ± 3.14	66.94 ± 3.52	56.18 ± 4.13
TENT W/ ODS	$\mathbf{69.00 \pm 5.96}$	$\mathbf{73.56 \pm 2.85}$	$\mathbf{66.03 \pm 1.89}$
COTTA	81.81 ± 0.37	73.58 ± 0.28	60.58 ± 0.15
COTTA W/ ODS	$\mathbf{82.11 \pm 0.25}$	$\mathbf{79.74 \pm 0.32}$	$\mathbf{74.72 \pm 0.64}$
NOTE	78.81 ± 0.27	77.96 ± 0.75	77.18 ± 0.38
NOTE W/ ODS	$\mathbf{81.13 \pm 0.09}$	$\mathbf{80.40 \pm 0.36}$	$\mathbf{80.67 \pm 0.29}$

RQ3: Does ODS accurately estimate label distribution and effectively optimize the prediction?

We conduct experiments on CIFAR10 dataset with $\gamma = 10$ to investigate the above question. First, we compare the estimation error of the three methods: 1) NOTE, the current SOTA robust TTA method; 2) LAME, a conservative robust TTA method; 3) ODS, our proposed framework combined with NOTE. For these three methods, we employ the same FIFO queue to compute mean label distribution as described in Section 3.1. We adopt KL-divergence (Joyce, 2011) to measure the error between ground-truth label distribution $\mathcal{D}_t(Y)$ and estimated label distribution $\boldsymbol{w}_t$ at each timestamp t. Figure 5 demonstrates that ODS can estimate label distribution more accurately than the other two methods at each timestamp t. Overall, ODS gives 0.066 KL-divergence on average, which is significantly better than 0.135 for NOTE and 0.134 for LAME. Then, we compare the confusion matrix of NOTE and ODS. Figure 6 and 7 show that ODS gives more accurate prediction for all classes compared to NOTE. This suggests that ODS framework can optimize the prediction effectively, aided by the estimated label distribution vector $\boldsymbol{w}_t$.

Table 5. Average results of three TTA methods w/ and w/o each module of ODS framework. The results show that the best results are achieved when combing both modules.

MODULES $\mathcal{M}_T$	MODULES $\mathcal{M}_O$	TENT	COTTA	NOTE
		56.18 ± 4.13	60.58 ± 0.15	77.18 ± 0.38
✓		58.95 ± 2.36	60.65 ± 0.31	77.20 ± 0.57
✓	✓	$\mathbf{66.03 \pm 1.89}$	$\mathbf{74.72 \pm 0.64}$	$\mathbf{80.67 \pm 0.29}$

4.3. Ablation Study

We investigate the contribution of each module of ODS on the CIFAR10 dataset with $\gamma = 10$. We combine ODS framework with three TTA methods, and the results are shown in Table 5. The first row gives the performance of the original TTA methods. Then, $\mathcal{M}_T$ is added to each method to track the label distribution and perform adaptively adaptation in the second row. Performance of TENT improves because $\mathcal{M}_T$ helps it tackle the imbalanced adaptation problem. COTTA and NOTE remain the same because they already adopt techniques for the non-i.i.d. data streams, e.g., anti-forgetting strategy and class-balanced data buffer. Finally, we show the performance of the entire ODS framework in the third row. $\mathcal{M}_O$ utilizes the label distribution estiamted by $\mathcal{M}_T$, and thereby enhancing the final performance. The results show that ODS significantly improves the performance of existing TTA methods, suggesting that the two modules, i.e., $\mathcal{M}_T$ and $\mathcal{M}_O$, are crucial to our framework.

4.4. More Discussion

Different implementations of $\mathcal{M}_O$ In Section 2.2, we first present a straightforward implementation for optimizing the predictions, which adjusts probabilities according to $\boldsymbol{w}_t$ in (2). Then, a conservative but robust implementation is proposed in Section 3.2 to get better results in practical applications. We denote these two implementations as Statistics Optimization (SO) and Distribution Optimization (DO). Table 6 gives the performance of two $\mathcal{M}_O$ implementations on CIFAR10C dataset with $\gamma = 10$. The results in the second column are consistent with the theoretical analysis in Section 2.2, and the third column proves that the implementation in Section 3.2 can give better results.

Table 6. Performance and wall-clock time of ODS with different $\mathcal{M}_O$ on CIFAR10 dataset.

	NOTE	ODS W/ SO	ODS W/ DO
PERFORMANCE	77.18 ± 0.38	79.50 ± 0.31	80.67 ± 0.29
AVG. TIME	0.1034S (100%)	0.1150S (111%)	0.1156S (112%)

Time Consumption. We analyze the time consumption of the ODS framework, using different implementations of $\mathcal{M}_O$. The last row of Table 6 reports the running time of each algorithm when making predictions on 1,900 images. The results show that ODS framework can significantly improve performance with only an additional 12% of time. Moreover, we analyze the time consumption of the COTTA method, which has good performance in our experiments. However, it is time-consuming due to the multiple augmentations during test time. Under the same setting, its average running time is 0.9148s. The results indicate that COTTA takes about 7 times longer than our proposed ODS framework. Therefore, these experiments prove that two versions of our proposed ODS framework are efficient and effective.

In-depth Comparison with LAME. LAME can revise the predictions of the model without adapting itself. A natural question is whether directly combining LAME with TTA methods can achieve similar improvement as our proposal. We compare ODS with the combination of LAME and NOTE on the CIFAR10 dataset. The combination can only improve the performance when $\gamma = 10$ while giving negative effects on other situations. ODS framework consistently improves the performance benefiting from $\mathcal{M}_T$ and $\mathcal{M}_O$ to estimate label distribution better and optimize the prediction, respectively.

Table 7. Comparison with ODS and combination of LAME and NOTE methods on CIFAR10 dataset. Bold indicates the best. Underline indicates degraded results. The direct combination does not always work, while ODS gives stable performance gains.

	NOTE	NOTE+LAME	ODS
γ=2	78.81 ± 0.27	$\underline{77.32 \pm 0.17}$	$\mathbf{81.13 \pm 0.09}$
γ=5	77.96 ± 0.75	$\underline{76.76 \pm 0.67}$	$\mathbf{80.40 \pm 0.36}$
γ=10	77.18 ± 0.38	$\underline{78.43 \pm 0.77}$	$\mathbf{80.67 \pm 0.29}$

5. Related Work

Test-time Adaptation. Test-time adaptation aims to adapt a source model to the distribution shift in testing data without using any source data. Test-time training studies, e.g, TTT (Sun et al., 2020), TTT+ (Liu et al., 2021), MT3 (Bartler et al., 2022), operate both model training and testing process. They additionally optimize self-supervised objectives at training time and adapt the model parameters via optimized self-supervised objectives at test time. However, these studies assume that the training phase is controllable, which limits the scope of applications. Fully test-time adaptation tackles the above limitation, adapting the model without assumptions on the source model. TENT (Wang et al., 2021) introduces entropy minimization to update the BN layers at test time. EATA (Niu et al., 2022) additionally proposes the sample selection and weighting strategies for efficiency. Other studies (Nado et al., 2020; Zhang et al., 2021; Goyal et al., 2022) also re-calibrate BN layers to ensure the performance at the testing phase. In practice, the deployed model continually works under non-i.i.d. scenarios. COTTA (Wang et al., 2022a) adopts the weight-averaged model, augmentation-averaged prediction, and stochastically restore to enable the continual adaptation ability in changing environments. LAME (Boudiaf et al., 2022) proposes a conservative approach, which revises the predictions without adapting the model. NOTE (Gong et al., 2022) adopts instance-aware batch normalization and prediction-balanced reservoir sampling to ensure robustness under non-i.i.d. scenarios. Some other studies (Niu et al., 2023; Yuan et al., 2023) also consider TTA in the practical scenarios. They play roles in promoting the deployment of the TTA method in practical applications. Our paper focuses on test-time adaptation settings where covariate and label distributions change together, and provides theoretical insights.

Distribution Shift. Covariate shift (Huang et al., 2006; Ruan et al., 2022; Zadrozny, 2004) assumes that the conditional distribution is constant, and marginal covariate distribution changes. In contrast, label shift (Alexandari et al., 2020; Azizzadenesheli et al., 2019; Zhao et al., 2021) studies cases where the class-conditional distribution remains the same, but the marginal label distribution changes. Some studies (des Combes et al., 2020; Luo & Ren, 2022) extend the label shift assumption to the generalized label shift assumption, which allows both covariate and label distributions to change together under specific constraints. Numerous studies tackle the above problems from the perspec-

tive of causal (Schölkopf et al., 2012), maximum mean discrepancy (Gretton et al., 2012), Wasserstein distance (Chen et al., 2018), etc. Our study adopts the generalized label shift assumption, but the difference is that existing studies (des Combes et al., 2020) mainly focus on handling the distribution shift offline with plenty of data, while our study adapts to the distribution shift at test time with limited data. Thus, our setting is practical yet challenging.

Online Label Shift. Learning from data streams (Zhou, 2023) cannot ignore that the incoming data stream can be potentially endless with unknown changes, e.g., label shift. Online Label shift studies the setting where the test-time label distribution continually changes, and the model should dynamically adapt to the shift without observing the true labels. One prior study (Wu et al., 2021) proposes adaptation algorithms based on classical online learning techniques. ATLAS (Bai et al., 2022) proposes an online ensemble algorithm (Zhao et al., 2022) for dealing with changing label distribution, which provides provable guarantees. However, these studies ignore the covariate shift, which usually exists in continually changing environments. Therefore, they can not effectively deal with the practical situation studied in this paper.

6. Conclusion

In this paper, we study the AODS problem where both covariate and label distributions change during the testing phase. Our goal is to adapt the source model to this open-world data shift. We address this problem by first theoretically analyzing the relationship between classification error and distribution shifts. Motivated by our analysis, we introduce a generic TTA framework ODS that tracks the dynamic label distribution while performing model adaptation adaptively and optimizing the prediction efficiently. ODS is generic to integrate with existing TTA methods and achieves consistent performance gains on benchmark datasets with varying degrees of label distribution shifts. Experiment results demonstrate the superior performance of ODS over the SOTA methods. Our work can motivate researchers in two directions. Adaptation to real-world distribution changes deserves further exploration due to its broad range of applications. In-depth theoretical studies are needed to help better design TTA algorithms.

One limitation of our framework is it cannot achieve the SOTA performance when no label distribution shift exists. Nevertheless, our ODS framework is still capable of achieving comparable performance in this scenario. In our future work, we will focus on designing a robust algorithm that delivers SOTA performance irrespective of label distribution. Moreover, it is also interesting to study how to incorporate the test-time adaptation to facilitate the generalization of the large pre-trained vision-language models.

Acknowledgements

This research was supported by the National Key R&D Program of China (2022ZD0114803), the National Science Foundation of China (62176118) and CAAI-Huawei MindSpore Open Fund.

References

Alexandari, A., Kundaje, A., and Shrikumar, A. Maximum likelihood with bias-corrected calibration is hard-to-beat at label shift adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 222–232, 2020.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 173–182, 2016.

Azizzadenesheli, K., Liu, A., Yang, F., and Anandkumar, A. Regularized learning for domain adaptation under label shifts. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Bai, Y., Zhang, Y.-J., Zhao, P., Sugiyama, M., and Zhou, Z.-H. Adapting to online label shift with provable guarantees. In *Advances in Neural Information Processing Systems*, 2022.

Bartler, A., Bühler, A., Wiewel, F., Döbler, M., and Yang, B. MT3: meta test-time training for self-supervised test-time adaption. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pp. 3080–3090, 2022.

Boudiaf, M., Mueller, R., Ben Ayed, I., and Bertinetto, L. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Chen, Q., Liu, Y., Wang, Z., Wassell, I. J., and Chetty, K. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7976–7985, 2018.

Chowdhury, G. G. Natural language processing. *Annual Review of Information Science and Technology*, 37(1): 51–89, 2003.

Collell, G., Prelec, D., and Patil, K. R. Reviving threshold-moving: a simple plug-in bagging ensemble for binary

and multiclass imbalanced data. *CoRR*, abs/1606.08698, 2016.

des Combes, R. T., Zhao, H., Wang, Y., and Gordon, G. J. Domain adaptation with conditional distribution matching and generalized label shift. In *Advances in Neural Information Processing Systems*, pp. 1645–1655, 2020.

Gong, T., Jeong, J., Kim, T., Kim, Y., Shin, J., and Lee, S.-J. NOTE: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*, pp. 27253–27266, 2022.

Goyal, S., Sun, M., Raghunathan, A., and Kolter, J. Z. Test time adaptation via conjugate pseudo-labels. In *Advances in Neural Information Processing Systems*, pp. 6204–6218, 2022.

Grandvalet, Y. and Bengio, Y. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pp. 529–536, 2004.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

Guo, J., Gong, M., Liu, T., Zhang, K., and Tao, D. LTF: A label transformation framework for correcting label shift. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3843–3853, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., and Schölkopf, B. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pp. 601–608, 2006.

Joyce, J. M. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*, pp. 720–722. Springer, 2011.

Kim, J., Hwang, I., and Kim, Y. M. Ev-tta: Test-time adaptation for event-based object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17724–17733, 2022.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Lipton, Z. C., Wang, Y., and Smola, A. J. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 3128–3136, 2018.

Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T., and Alahi, A. TTT++: When does self-supervised test-time training fail or thrive? In *Advances in Neural Information Processing Systems*, pp. 21808–21820, 2021.

Luo, Y. and Ren, C. Generalized label shift correction via minimum uncertainty principle: Theory and algorithm. *CoRR*, abs/2202.13043, 2022.

Martinez, J., Hossain, R., Romero, J., and Little, J. J. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2659–2668, 2017.

Menon, A. K., Narasimhan, H., Agarwal, S., and Chawla, S. On the statistical consistency of algorithms for binary classification under class imbalance. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 603–611, 2013.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, abs/2006.10963, 2020.

Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16888–16905, 2022.

Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., and Tan, M. Towards stable test-time adaptation in dynamic wild world. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

Ruan, Y., Dubois, Y., and Maddison, C. J. Optimal representations for covariate shift. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against

common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems*, pp. 11539–11551, 2020.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. M. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Shin, I., Tsai, Y., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I. S., and Yoon, K. MM-TTA: multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16907–16916, 2022.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9229–9248, 2020.

Tasche, D. Fisher consistency for prior probability shift. *Journal of Machine Learning Research*, 18:95:1–95:32, 2017.

Wagner, T., Guha, S., Kasiviswanathan, S. P., and Mishra, N. Semi-supervised learning on data streams via temporal label propagation. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5082–5091, 2018.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B. A., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

Wang, Q., Fink, O., Gool, L. V., and Dai, D. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7191–7201, 2022a.

Wang, Y., Chen, H., Fan, Y., Sun, W., Tao, R., Hou, W., Wang, R., Yang, L., Zhou, Z., Guo, L.-Z., Qi, H., Wu, Z., Li, Y.-F., Nakamura, S., Ye, W., Savvides, M., Raj, B., Shinozaki, T., Schiele, B., Wang, J., Xie, X., and Zhang, Y. Usb: A unified semi-supervised learning benchmark for classification. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2022b.

Wu, R., Guo, C., Su, Y., and Weinberger, K. Q. Online adaptation to label distribution shift. In *Advances in Neural Information Processing Systems*, pp. 11340–11351, 2021.

Yang, X., Sun, H., Sun, X., Yan, M., Guo, Z., and Fu, K. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access*, 6:50839–50849, 2018.

Yuan, L., Xie, B., and Li, S. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Yuille, A. L. and Rangarajan, A. The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, pp. 1033–1040, 2001.

Zadrozny, B. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21st International Conference on Machine Learning*, volume 69, 2004.

Zhang, M. M., Levine, S., and Finn, C. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Zhao, E., Liu, A., Anandkumar, A., and Yue, Y. Active learning under label shift. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 3412–3420, 2021.

Zhao, P., Xie, Y.-F., Zhang, L., and Zhou, Z.-H. Efficient methods for non-stationary online learning. In *Advances in Neural Information Processing Systems*, pp. 11573–11585, 2022.

Zhou, Z.-H. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

Zhou, Z.-H. *Ensemble learning*. Springer, 2021.

Zhou, Z.-H. Stream efficient learning. *CoRR*, abs/2305.02217, 2023.

A. Proof of Theorem 2.3

To simplify the notations, we define $\gamma_0 = \mathcal{D}_0(Y)$ and $\gamma_t = \mathcal{D}_t(Y)$, representing the label distribution at each timestamp. Note that we assume that source data distribution is class-balanced. Therefore, γ_0 also represents a uniform distribution γ.

Proof. First, we derive the following equality based on the law of total probability:

$$\begin{aligned}
&\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right| \\
=&\left|\mathcal{D}_0(\hat{Y} \neq Y) - \mathcal{D}_0(\hat{Y}_o \neq Y)\right| \\
=&\left|\sum_{i\neq j} \mathcal{D}_0(\hat{Y} = i, Y = j) - \sum_{i\neq j} \mathcal{D}_0(\hat{Y}_0 = i, Y = j)\right| \\
=&\left|\sum_{i\neq j} \gamma_{0,j}\mathcal{D}_0(\hat{Y} = i|Y = j) - \sum_{i\neq j} \gamma_{t,j}\mathcal{D}_t(\hat{Y}_o = i|Y = j)\right|
\end{aligned} \tag{A.1}$$

Then, we can bound the above error gap:

$$\begin{aligned}
&\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right| \\
=&\left|\sum_{i\neq j} \gamma_{0,j}\mathcal{D}_0(\hat{Y} = i|Y = j) - \sum_{i\neq j} \gamma_{t,j}\mathcal{D}_t(\hat{Y}_o = i|Y = j)\right| \\
\leq&\sum_{i\neq j}\left|\gamma_{0,j}\mathcal{D}_0(\hat{Y} = i|Y = j) - \gamma_{t,j}\mathcal{D}_t(\hat{Y}_o = i|Y = j)\right|
\end{aligned} \tag{A.2}$$

According to the (8) in (Menon et al., 2021), $\hat{Y}_0$ gives the following equation:

$$\begin{aligned}
\mathcal{D}_t(\hat{Y}_o = j|Z) &= \frac{w_{t,j}}{\gamma_j}\mathcal{D}_t(\hat{Y} = j|Z) \\
\mathcal{D}_t(\hat{Y}_o = j|Z)\mathcal{D}_t(Z|Y) &= \frac{w_{t,j}}{\gamma_j}\mathcal{D}_t(\hat{Y} = j|Z)\mathcal{D}_t(Z|Y) \\
\mathcal{D}_t(\hat{Y}_o = j|Y) &= \frac{w_{t,j}}{\gamma_j}\mathcal{D}_t(\hat{Y} = j|Y)
\end{aligned} \tag{A.3}$$

Combining (A.2) and (A.3), we can bound the error gap by:

$$\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right| \leq \sum_{i\neq j}\left|\gamma_{0,j}\mathcal{D}_0(\hat{Y} = i|Y = j) - \gamma_j\frac{\gamma_{t,j}}{w_{t,j}}\mathcal{D}_t(\hat{Y} = i|Y = j)\right| \tag{A.4}$$

Invoking Lemma A.2 in (des Combes et al., 2020) to bound the above term, we have:

$$\begin{aligned}
&\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right| \\
\leq&\sum_{i\neq j}\gamma_j|1 - \frac{\gamma_{t,j}}{w_{t,j}}| \cdot \Big(\alpha_j\mathcal{D}_0(\hat{Y} = i|Y = j) + \beta_j\mathcal{D}_t(\hat{Y} = i|Y = j)\Big) + \gamma_{0,j}\Delta_{CE}(\hat{Y}) + \gamma_j\frac{\gamma_{t,j}}{w_{t,j}}\Delta_{CE}(\hat{Y}) \\
=&\sum_{i\neq j}\gamma_j|1 - \frac{\gamma_{t,j}}{w_{t,j}}| \cdot \Big(\alpha_j\mathcal{D}_0(\hat{Y} = i|Y = j) + \beta_j\mathcal{D}_t(\hat{Y} = i|Y = j)\Big) + 2(K-1)\Delta_{CE}(\hat{Y})
\end{aligned} \tag{A.5}$$

where α_j, β_j are some non-negative constants satisfying $\alpha_j + \beta_j = 1$. We can set $\alpha_j = 1, \beta_j = 0, \forall j \in [K]$ and then use Holder's inequality:

$$\begin{aligned}
&\left|\varepsilon_0(\hat{Y}) - \varepsilon_t(\hat{Y}_o)\right| \\
\leq&\sum_{i\neq j}\gamma_j|1 - \frac{\gamma_{t,j}}{w_{t,j}}| \cdot \mathcal{D}_0(\hat{Y} = i|Y = j) + 2(K-1)\Delta_{CE}(\hat{Y}) \\
\leq&C\|\mathbf{1} - \frac{\gamma_t}{\boldsymbol{w}_t}\|_1 \cdot BSE(\hat{Y}) + 2(K-1)\Delta_{CE}(\hat{Y})
\end{aligned} \tag{A.6}$$

Table 8. Performance on CIFAR10 dataset with no label distribution shifts. Bold indicates best.

Methods	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
Source	24.27 ± 0.00	30.36 ± 0.00	20.66 ± 0.00	57.62 ± 0.00	47.19 ± 0.00	65.50 ± 0.00	64.41 ± 0.00	77.19 ± 0.00	62.92 ± 0.00	71.31 ± 0.00	91.11 ± 0.00	35.23 ± 0.00	76.22 ± 0.00	49.65 ± 0.00	74.30 ± 0.00	56.53
BN Stats	69.28 ± 0.25	71.82 ± 0.05	62.29 ± 0.24	87.39 ± 0.15	66.48 ± 0.13	85.65 ± 0.03	87.19 ± 0.17	81.96 ± 0.13	81.13 ± 0.16	85.10 ± 0.09	91.01 ± 0.12	86.03 ± 0.11	77.14 ± 0.16	79.66 ± 0.32	72.62 ± 0.28	78.98
Tent	74.78 ± 0.70	77.10 ± 0.93	63.86 ± 2.04	72.16 ± 3.31	53.72 ± 2.24	58.05 ± 4.67	57.79 ± 5.75	53.05 ± 4.57	49.17 ± 5.64	47.51 ± 5.94	47.49 ± 6.61	40.02 ± 7.91	34.40 ± 8.19	35.06 ± 10.95	30.92 ± 10.14	53.01
Eata	73.77 ± 0.45	76.75 ± 0.55	65.61 ± 0.47	87.42 ± 0.24	67.84 ± 0.51	85.52 ± 0.21	87.53 ± 0.13	82.99 ± 0.27	82.11 ± 0.28	85.47 ± 0.32	91.16 ± 0.15	85.95 ± 0.20	77.57 ± 0.30	81.23 ± 0.47	74.71 ± 0.45	80.38
Lame	69.51 ± 0.23	72.07 ± 0.22	62.12 ± 0.18	87.60 ± 0.10	66.35 ± 0.07	86.04 ± 0.05	87.51 ± 0.12	82.26 ± 0.17	81.46 ± 0.17	85.41 ± 0.04	91.31 ± 0.13	86.24 ± 0.12	77.50 ± 0.19	80.02 ± 0.20	72.85 ± 0.10	79.22
CoTTA	**77.54 ± 0.19**	**80.17 ± 0.33**	**75.44 ± 0.41**	87.55 ± 0.44	**74.77 ± 0.19**	85.41 ± 0.24	87.01 ± 0.29	83.11 ± 0.20	83.24 ± 0.36	84.34 ± 0.35	88.74 ± 0.37	83.84 ± 0.37	80.00 ± 0.45	83.42 ± 0.21	**80.53 ± 0.36**	**82.34**
Note	65.19 ± 0.78	77.98 ± 0.32	68.34 ± 0.81	78.51 ± 1.01	67.58 ± 0.50	85.60 ± 0.15	87.75 ± 0.55	85.08 ± 0.19	85.84 ± 0.54	84.56 ± 0.59	91.55 ± 0.19	90.82 ± 0.42	79.32 ± 0.39	75.82 ± 1.07	76.65 ± 0.83	80.04
ODS+ Tent	74.56 ± 0.53	77.60 ± 1.90	66.00 ± 3.49	77.52 ± 5.45	59.57 ± 6.25	67.31 ± 9.00	68.38 ± 10.39	64.65 ± 9.80	60.14 ± 8.52	58.15 ± 8.57	59.25 ± 11.24	47.64 ± 10.54	46.01 ± 8.72	45.18 ± 9.24	39.67 ± 8.18	60.77
ODS+ CoTTA	73.88 ± 0.55	77.15 ± 0.22	67.31 ± 0.65	**88.64 ± 0.16**	70.89 ± 0.21	87.38 ± 0.05	**88.90 ± 0.11**	84.32 ± 0.20	84.24 ± 0.26	**86.76 ± 0.11**	91.81 ± 0.16	87.90 ± 0.26	**80.78 ± 0.23**	**83.89 ± 0.13**	78.08 ± 0.17	82.13
ODS+ Note	69.23 ± 0.44	77.47 ± 0.36	68.24 ± 0.40	84.60 ± 0.26	69.00 ± 0.40	**87.91 ± 0.26**	88.72 ± 0.25	**86.36 ± 0.19**	**87.17 ± 0.13**	86.33 ± 0.29	**92.18 ± 0.18**	**92.54 ± 0.12**	80.54 ± 0.35	80.32 ± 0.81	77.38 ± 0.78	81.87

Table 9. Performance on the CIFAR100 dataset with no label distribution shifts. Bold indicates best.

Methods	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
Source	11.88 ± 0.00	13.91 ± 0.00	6.61 ± 0.00	32.74 ± 0.00	21.58 ± 0.00	42.45 ± 0.00	40.24 ± 0.00	45.04 ± 0.00	33.34 ± 0.00	40.01 ± 0.00	65.93 ± 0.00	17.43 ± 0.00	51.37 ± 0.00	23.92 ± 0.00	44.30 ± 0.00	32.72
BN Stats	39.81 ± 0.17	40.66 ± 0.17	33.84 ± 0.22	64.18 ± 0.25	41.73 ± 0.27	62.01 ± 0.16	64.16 ± 0.28	52.69 ± 0.21	53.16 ± 0.11	56.60 ± 0.21	67.24 ± 0.13	61.32 ± 0.11	53.15 ± 0.33	56.07 ± 0.35	43.46 ± 0.06	52.67
Tent	**49.27 ± 0.75**	50.37 ± 0.93	38.18 ± 1.19	42.94 ± 2.42	28.46 ± 2.54	28.47 ± 3.08	25.68 ± 4.68	16.30 ± 4.17	10.91 ± 3.42	7.57 ± 2.43	6.37 ± 2.32	3.35 ± 0.97	3.73 ± 0.81	3.36 ± 0.64	2.95 ± 0.52	21.19
Eata	41.82 ± 0.80	36.55 ± 13.90	27.68 ± 12.85	50.80 ± 24.57	32.58 ± 15.42	49.00 ± 23.67	50.80 ± 24.47	42.42 ± 20.31	42.22 ± 20.24	45.71 ± 22.21	53.26 ± 25.92	48.99 ± 23.99	41.40 ± 20.18	45.49 ± 22.26	35.88 ± 17.43	42.97
Lame	38.35 ± 0.38	39.23 ± 1.06	31.67 ± 0.61	64.54 ± 0.29	40.83 ± 0.33	62.41 ± 0.18	64.48 ± 0.28	52.74 ± 0.33	53.08 ± 0.37	56.44 ± 0.23	67.82 ± 0.09	61.69 ± 0.21	53.26 ± 0.39	56.47 ± 0.29	42.75 ± 0.18	52.39
CoTTA	45.80 ± 0.85	45.74 ± 1.15	41.04 ± 0.98	52.69 ± 0.89	41.86 ± 0.98	47.82 ± 1.15	48.09 ± 0.98	42.27 ± 1.04	41.77 ± 1.14	39.00 ± 1.04	47.06 ± 1.02	39.83 ± 0.87	40.33 ± 0.87	42.31 ± 1.08	39.72 ± 0.90	43.69
Note	35.99 ± 0.79	49.45 ± 0.55	40.82 ± 0.19	51.85 ± 1.32	41.81 ± 0.42	60.46 ± 0.81	62.59 ± 0.38	56.57 ± 0.24	59.76 ± 0.21	54.55 ± 0.77	64.71 ± 0.63	60.88 ± 0.51	51.79 ± 0.42	48.67 ± 2.07	48.03 ± 0.76	52.53
ODS+ Tent	48.16 ± 0.30	**52.72 ± 0.48**	**42.54 ± 0.40**	57.91 ± 0.77	37.13 ± 1.28	44.48 ± 1.90	40.16 ± 2.78	26.30 ± 2.54	19.56 ± 2.86	15.82 ± 2.75	17.07 ± 3.21	9.76 ± 1.24	8.04 ± 0.80	7.68 ± 0.43	5.00 ± 0.48	28.82
ODS+ CoTTA	44.23 ± 0.35	46.49 ± 0.42	40.84 ± 0.54	**65.21 ± 0.37**	**47.11 ± 0.43**	63.25 ± 0.21	64.76 ± 0.17	55.55 ± 0.36	55.64 ± 0.39	57.56 ± 0.18	67.49 ± 0.23	61.36 ± 0.08	55.87 ± 0.33	**58.91 ± 0.18**	48.99 ± 0.31	55.55
ODS+ Note	44.25 ± 0.45	49.11 ± 0.40	40.97 ± 0.43	64.54 ± 0.28	46.02 ± 0.27	**65.95 ± 0.19**	**65.43 ± 0.10**	**60.13 ± 0.32**	**61.68 ± 0.26**	**59.38 ± 0.28**	**70.87 ± 0.07**	**69.56 ± 0.19**	**57.89 ± 0.12**	58.55 ± 0.76	**51.11 ± 0.46**	**57.70**

where C is one constant related the $\mathcal{D}_0(Y)$. □

B. Experimental Details

All experiments are repeatedly conducted with one NVIDIA GeForce RTX 3090 GPU with a random seed setting from 0 to 4. In this section, we introduce baseline implementations and dataset details as follows.

For all comparison methods, we referred to their official implementation and reported hyperparameters in their original paper. If the hyperparameters on the corresponding dataset are not provided for one method, we will further tune the hyperparameters for it. Following previous study (Wang et al., 2022a), each method is optimized by Adam Optimizer (Kingma & Ba, 2015) if not specifically stated. For ODS, we adopt the same hyperparameters as LAME for calculating the similarity between samples. The details are shown as follows:

- TENT (Wang et al., 2021) sets the learning rate to 0.001 for all datasets. All experiments about TENT are implemented based on their official code[1].
- EATA (Niu et al., 2022) sets the learning rate to 0.005 for all datasets. The entropy constant E_0 is set to $0.4 \ln 10$ for CIFAR10 dataset and $0.4 \ln 100$ for CIFAR100 datases. The threshold ϵ is set to 0.4 for CIFAR10 and CIFAR100 datasets. In the experiments, we adopt source data for calculating weight importance to measure the upper bound of its performance. Implementation is referred to official code[2].
- LAME (Boudiaf et al., 2022) adopts the RBF kernel to calculate the similarity between samples. For the CIFAR10 dataset, the KNN hyperparameter is set to 5. For the CIFAR100 dataset, the KNN hyperparameter is set to 2. Imple-

[1]https://github.com/DequanWang/tent
[2]https://github.com/mr-eggplant/EATA

Table 10. Detailed results of Table 1

Methods	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
Source	14.70 ± 0.28	18.52 ± 0.27	15.61 ± 0.66	56.92 ± 0.75	31.99 ± 0.81	68.01 ± 0.67	63.25 ± 0.83	82.19 ± 0.53	72.44 ± 0.31	76.31 ± 0.43	**92.41 ± 0.70**	23.38 ± 0.45	72.33 ± 0.61	68.72 ± 0.63	79.72 ± 0.36	55.77
BN Stats	50.60 ± 0.65	51.16 ± 0.36	45.31 ± 0.47	71.73 ± 0.68	47.99 ± 0.58	69.35 ± 0.56	68.59 ± 0.62	60.16 ± 0.71	60.39 ± 0.55	64.27 ± 0.40	69.60 ± 0.62	67.56 ± 0.71	59.21 ± 0.47	66.12 ± 0.47	58.17 ± 0.38	60.68
Tent	53.53 ± 1.45	60.97 ± 1.29	59.34 ± 2.15	63.33 ± 2.53	47.12 ± 4.22	65.81 ± 5.37	68.11 ± 6.84	55.08 ± 3.88	55.00 ± 4.83	58.68 ± 5.35	63.40 ± 7.06	49.59 ± 4.28	46.95 ± 5.94	50.45 ± 9.98	45.38 ± 10.49	56.18
Eata	48.94 ± 0.60	48.21 ± 0.88	42.05 ± 0.79	65.44 ± 1.54	43.42 ± 1.73	59.81 ± 1.20	57.27 ± 1.01	55.09 ± 1.60	52.98 ± 0.99	56.00 ± 1.32	59.54 ± 0.92	61.47 ± 1.83	51.32 ± 0.35	55.75 ± 1.22	50.88 ± 2.23	53.88
Lame	57.99 ± 0.67	60.15 ± 1.28	53.07 ± 1.28	78.83 ± 0.68	53.04 ± 0.66	76.67 ± 1.04	74.90 ± 0.20	67.81 ± 1.23	67.30 ± 0.77	71.94 ± 0.50	77.05 ± 0.90	74.84 ± 0.86	68.53 ± 0.98	73.44 ± 0.78	66.90 ± 0.23	68.16
CoTTA	57.43 ± 0.66	60.06 ± 0.32	56.03 ± 0.73	66.66 ± 0.64	52.25 ± 0.73	66.54 ± 0.58	66.65 ± 0.78	58.32 ± 0.79	58.92 ± 1.10	60.09 ± 0.97	64.69 ± 1.29	55.05 ± 1.50	59.37 ± 0.68	64.74 ± 0.35	61.92 ± 1.00	60.58
Note	51.90 ± 0.91	54.57 ± 2.54	68.38 ± 1.33	84.29 ± 1.14	50.53 ± 2.11	88.97 ± 0.84	86.21 ± 1.35	86.15 ± 0.98	86.68 ± 1.08	83.27 ± 1.12	86.48 ± 1.51	90.64 ± 0.49	77.84 ± 1.51	80.77 ± 1.09	81.02 ± 1.10	77.18
ODS+ Tent	58.96 ± 0.79	64.11 ± 0.66	60.20 ± 1.52	73.71 ± 1.22	49.87 ± 3.72	70.86 ± 3.50	69.74 ± 3.25	67.21 ± 4.27	68.04 ± 4.82	70.81 ± 4.69	76.57 ± 5.37	67.14 ± 2.81	62.42 ± 3.70	68.38 ± 4.36	62.46 ± 5.17	66.03
ODS+ CoTTA	66.56 ± 1.55	**68.99 ± 1.92**	64.16 ± 1.42	82.29 ± 0.49	**59.18 ± 2.81**	80.58 ± 1.18	78.81 ± 0.83	77.34 ± 0.76	75.24 ± 1.55	77.18 ± 1.01	83.78 ± 0.71	76.48 ± 0.96	75.94 ± 0.43	79.64 ± 0.27	74.60 ± 0.56	74.72
ODS+ Note	**67.45 ± 1.91**	65.78 ± 2.78	**71.88 ± 1.46**	**88.66 ± 0.64**	56.32 ± 1.67	**90.48 ± 0.38**	**88.09 ± 0.82**	**86.16 ± 0.35**	**86.93 ± 1.11**	**83.96 ± 0.82**	87.37 ± 0.58	**91.16 ± 0.54**	**79.35 ± 1.27**	**84.43 ± 0.57**	**82.02 ± 0.97**	**80.67**

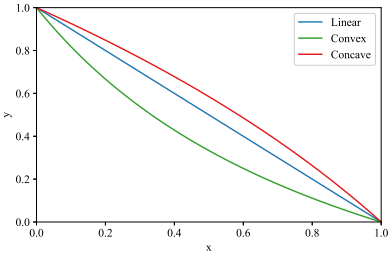


Figure 8. Illustration of different functions.

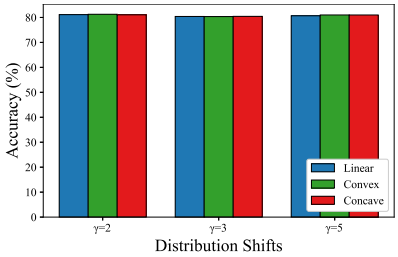


Figure 9. Comparison of different functions.

mentation is referred to official code[3].

- CoTTA (Wang et al., 2022a) sets the learning rate to 0.001 for all datasets. The restoration probability is set to 0.01. The augmentation threshold p_{th} is set to 0.72 for the CIFAR100 dataset and 0.92 for the CIFAR10 dataset. Same as in the original paper, we use 32 augmentations for our experiments. More augmentations can bring a slight performance improvement, but it will seriously affect the prediction speed. Implementation is referred to official code[4].
- Note: (Gong et al., 2022) sets the learning rate to 0.0001 and the queue size to 64 for all datasets. Implementation is referred to official code[5].

For datasets, we adopted the tweak-one shift introduced in (Guo et al., 2020) to simulate the changing label distribution in the testing phase. CIFAR10-C and CIFAR100-C datasets contain 15 corruptions falling into four categories: Noise, Blur, Weather, and Digital. We selected one or several classes for each category and increase their probabilities. For the CIFAR10 dataset, we selected classes 0, 1, 8, and 9 for the above four categories, respectively. For the CIFAR100 dataset, we selected super-class pairs (0, 1), (5, 6), (9, 10), and (18, 19) for the above four categories. Samples belonging to selected classes will be γ times more likely to occur than the other samples during the corresponding corruptions.

C. Additional Experimental Results

C.1. Discussion about $S(\cdot)$

We utilize a linear function $S(\boldsymbol{w}_t) = \text{Normalize}(\mathbf{1} - \boldsymbol{w}_t)$ to transform $\boldsymbol{w}_t$ into adaptive weights for (5). In order to further explore the influence of S, we choose three different functions for experiments: $S_{Linear}(\boldsymbol{w}_t) = \text{Normalize}(\mathbf{1} - \boldsymbol{w}_t)$, $S_{Convex}(\boldsymbol{w}_t) = \text{Normalize}(\frac{\mathbf{1}-\boldsymbol{w}_t}{\mathbf{1}-\boldsymbol{w}_t})$ and $S_{Concave}(\boldsymbol{w}_t) = \text{Normalize}(\frac{\ln \mathbf{2}-\boldsymbol{w}_t}{\ln 2})$. Illustration of different functions is shown

[3]https://github.com/fiveai/LAME
[4]https://github.com/qinenergy/cotta
[5]https://github.com/TaesikGong/NOTE

in Figure 8. The results in Figure 9 show that the choice of specific functions does not significantly impact the performance, and they all can effectively realize the adaptive adaptation. At the same time, in order to prevent numerical errors, we add $\epsilon = 0.1$ to the denominator uniformly. Therefore, we use the simple linear function in the original paper.

C.2. Detailed Results

For space reasons, we only present the results on the CIFAR10 dataset with $\gamma = 10$ and omit the standard deviation term in the paper. Here we first give detailed results in Table 10. Then, we present the performance with no label distribution shifts in Table 8. Since there is no change in label distribution, the best average performance is obtained from existing TTA methods. In this case, our algorithm is also able to adaptively track the unchanged and balanced label distribution and gives competitive results. This proves that our method can obtain robust and excellent results in various situations.