Towards Real-World Test-Time Adaptation: Tri-net Self-Training with Balanced Normalization

Yongyi Su[1], **Xun Xu**[2, 1*], **Kui Jia**[3*]
[1]South China University of Technology
[2]Institute for Infocomm Research, A*STAR
[3]School of Data Science, The Chinese University of Hong Kong, Shenzhen
eesuyongyi@mail.scut.edu.cn, alex.xun.xu@gmail.com, kuijia@cuhk.edu.cn

Abstract

Test-Time Adaptation aims to adapt source domain model to testing data at inference stage with success demonstrated in adapting to unseen corruptions. However, these attempts may fail under more challenging real-world scenarios. Existing works mainly consider real-world test-time adaptation under non-i.i.d. data stream and continual domain shift. In this work, we first complement the existing real-world TTA protocol with a globally class imbalanced testing set. We demonstrate that combining all settings together poses new challenges to existing methods. We argue the failure of state-of-the-art methods is first caused by indiscriminately adapting normalization layers to imbalanced testing data. To remedy this shortcoming, we propose a balanced batchnorm layer to swap out the regular batchnorm at inference stage. The new batchnorm layer is capable of adapting without biasing towards majority classes. We are further inspired by the success of self-training (ST) in learning from unlabeled data and adapt ST for test-time adaptation. However, ST alone is prone to over adaption which is responsible for the poor performance under continual domain shift. Hence, we propose to improve self-training under continual domain shift by regularizing model updates with an anchored loss. The final TTA model, termed as TRIBE, is built upon a tri-net architecture with balanced batchnorm layers. We evaluate TRIBE on four datasets representing real-world TTA settings. TRIBE consistently achieves the state-of-the-art performance across multiple evaluation protocols. The code is available at https://github.com/Gorilla-Lab-SCUT/TRIBE.

Introduction

The recent success of deep neural networks relies on the assumption of generalizing pre-trained model to i.i.d. testing domain (Wang et al. 2022a). When deep learning models are to be deployed on real-world applications, robustness to out-of-distribution testing data, e.g. visual corruptions caused by lighting conditions, adverse weather, etc. becomes a major concern. Recent studies revealed such corruptions could severely deteriorate the generalization of model pre-trained on clean training samples (Sun et al. 2022; Hendrycks and Dietterich 2019; Sakaridis, Dai, and Van Gool 2018). Importantly, the corruption on testing data is often unknown and sometimes unpredictable before deployment. Therefore, a new line of works emerge by adapting pre-trained models to testing data distribution at inference stage, a.k.a. test-time adaptation (TTA) (Sun et al. 2020; Wang et al. 2021; Su, Xu, and Jia 2022). The success of test-time adaptation is often achieved by distribution alignment (Su, Xu, and Jia 2022; Liu et al. 2021), self-supervised training (Chen et al. 2022) and self-training (Goyal et al. 2022), all demonstrating remarkable improvement of robustness on multiple types of visual corruptions in the testing data. Despite the unprecedented performance, existing TTA approaches are often developed under restrictive assumptions of testing data, e.g. stationary class distribution and static domain shift, and this gives rise to many attempts to explore TTA methods for real-world testing data (Wang et al. 2022b; Yuan, Xie, and Li 2023; Gong et al. 2022; Niu et al. 2023).

The recently explored real-world TTA, a.k.a. wild TTA (Niu et al. 2023) or Practical TTA (Yuan, Xie, and Li 2023), settings mainly consider the challenges brought by local class-imbalance (Niu et al. 2023; Yuan, Xie, and Li 2023; Gong et al. 2022) and continual domain shift (Wang et al. 2022b) which are expected to be encountered in real-world applications. Local class-imbalance is often observed when testing data are drawn in a non-i.i.d. manner (Gong et al. 2022). Direct adaptation indiscriminately results in biased distribution estimation and the recent works proposed exponential batchnorm update (Yuan, Xie, and Li 2023) or instance batchnorm update (Gong et al. 2022) to tackle this challenge. In this work, our aim is to address beyond the local class-imbalance challenge by taking into account the fact that the global distribution of testing data could be severely imbalanced and the class distribution may shift over time. We provide an illustration of the more challenging scenario in Fig. 1. This additional challenge renders existing TTA methods ineffective as the class prevalence on testing data is unknown before inference stage and the model could be biased towards majority classes through blind test-time adaptation. Through empirical observations, this issue becomes particularly acute for methods relying on estimating global statistics for updating normalization layers(Nado et al. 2020; Lee et al. 2013; Wang et al. 2021). It mainly owes to the fact that a single global distribution is estimated from the whole testing data on which samples are normalized. As such, the global distribution could easily bias towards majority classes,

*Correspondence to alex.xun.xu@gmail.com and kuijia@cuhk.edu.cn.

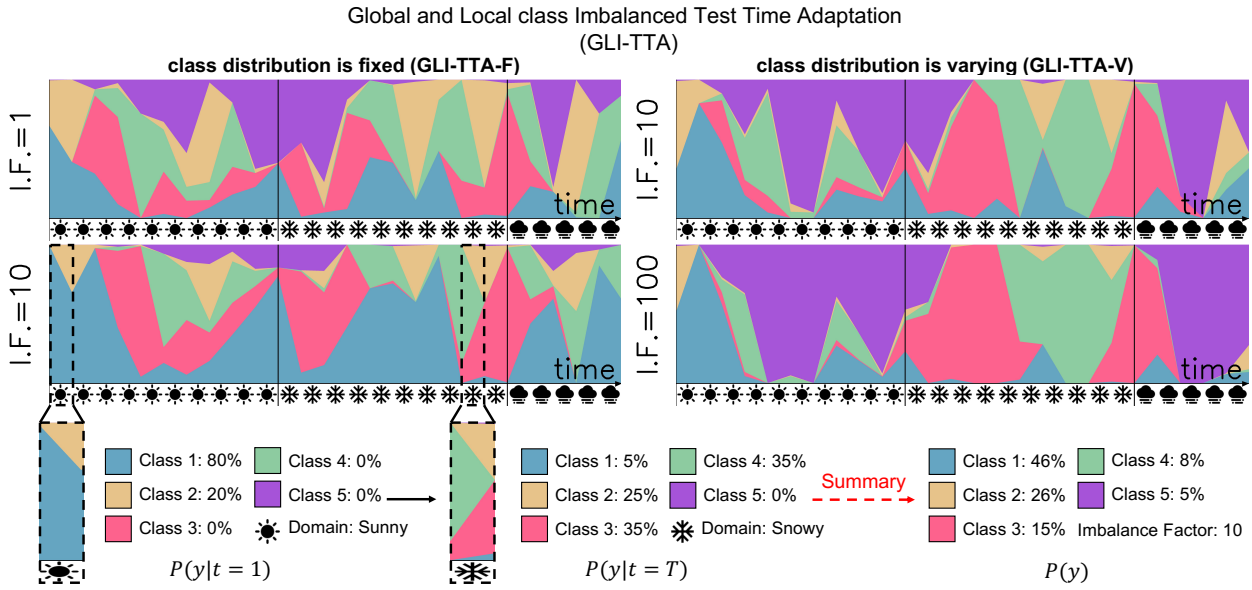

Figure 1: Illustration of two challenging real-world TTA scenarios. Different colors indicate the proportions of semantic classes, horizontal axis indicates testing data domain (e.g. different corruptions) may shift over time and different imbalance factor ($I.F.$) controls the degree of global imbalance. We expect the testing data stream to exhibit both local and global class imbalance, termed as "class distribution is fixed (GLI-TTA-F)" and this distribution may also evolve over time, termed as "class distribution is varying (GLI-TTA-V)".

resulting in internal covariate shift (Ioffe and Szegedy 2015). To avoid biased batch normalization (BN), we propose a balanced batch normalization layer by modeling the distribution for each individual category and the global distribution is extracted from category-wise distributions. The balanced BN allows invariant estimation of distribution under both locally and globally class-imbalanced testing data.

Shift of domain over time occurs frequently in real-world testing data, e.g. a gradual change of lighting/weather conditions. It poses another challenge to existing TTA methods as the model could overly adapt to domain A and struggle with domain B when A shifts to B. To alleviate overly adapting to a certain domain, CoTTA (Wang et al. 2022b) randomly reverts model weights to pre-trained weights and EATA (Niu et al. 2022) regularizes the adapted model weights against source pre-trained weights to avoid overly shifting model weights. Nevertheless, these approaches still do not explicitly address the challenge of constant shifting domains in testing data. As self-training has been demonstrated to be effective for learning from unlabeled data (Sohn et al. 2020), we adopt a teacher-student framework for TTA. Nonetheless, direct self-training without regularization is prone to confirmation bias (Arazo et al. 2020) and could easily overly adapt pre-trained model to a certain domain, causing degenerate performance upon seeing new domains. To avoid this over adaptation, we further introduce an anchor network, of which the weights are copied from pre-trained model and batch-norm layers are dynamically updated by testing samples. The anchored loss, realised as mean square error (MSE), between teacher and anchor network is jointly optimised with self-training loss to strike a balance between adaptation to specific domain and being versatile on ever changing domains. We brand this design as a tri-net architecture. We demonstrate that with the help of tri-net, TTA maintains a good performance within a wider range of learning rate. We refer to the final model as **TRI**-net self training with **B**alanc**E**d normalization (**TRIBE**) in recognition of the tri-net architecture with balanced normalization layer.

We summarize the contributions of this work as follows.

- We are motivated by the challenges in real-world test-time adaptation and propose to tackle a challenging TTA setting where testing data is both locally and globally class-imbalanced and testing domain may shift over time.
- A novel balanced batch normalization layer is introduced to fit to testing data distribution with both local and global class imbalance.
- We further introduce a tri-net framework to facilitate adaptation under continually shifting testing domain. We demonstrate this tri-net design improves robustness to the choice of learning rate.
- We evaluate the proposed method, TRIBE, on four test-time adaptation datasets under different real-world scenarios, demonstrating superior performance to all state-of-the-art methods.

Related Work

Unsupervised Domain Adaptation: Machine learning models often assume both training and testing data are drawn i.i.d. from the same distribution. When such assumption is violated, generalizing source model to testing distribution is

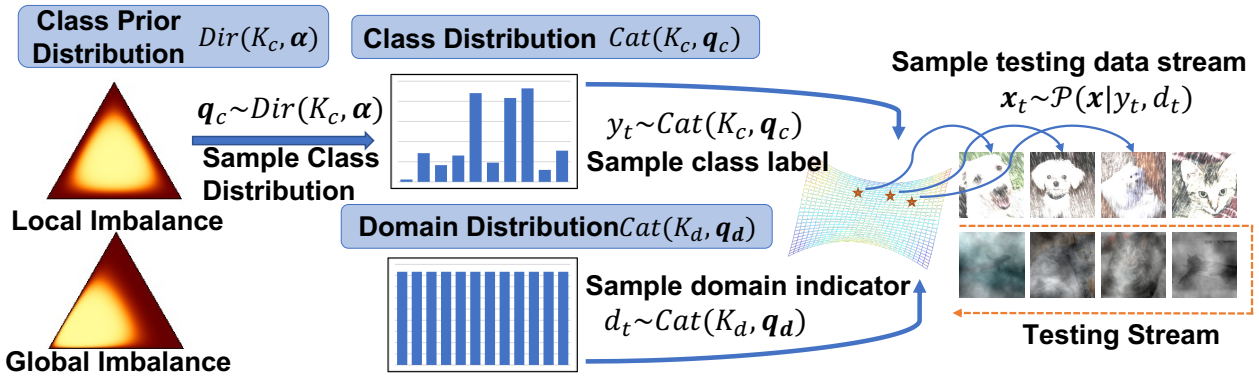

Figure 2: An illustration of the proposed real-world TTA simulation protocol with a hierarchical probabilistic model. A non-uniform α results in globally imbalanced testing data distribution.

hampered by the domain shift, leading to degraded performance (Wang and Deng 2018). Unsupervised domain adaptation (UDA) improves model generalization by exploiting both labeled source domain data and unlabeled target domain data (Ganin and Lempitsky 2015; Tzeng et al. 2014; Long et al. 2015). Common approaches towards UDA includes distribution alignment (Gretton et al. 2012; Sun and Saenko 2016; Zellinger et al. 2016), adversarial learning (Hoffman et al. 2018), target clustering (Tang, Chen, and Jia 2020) and self-training (Liu, Wang, and Long 2021). Nevertheless, UDA is only effective when source and target domain data are simultaneously accessible. More importantly, in real-world applications the distribution in target domain is often predictable until inference stage which has motivated research into test-time adaptation.

Test-Time Adaptation: Adapting pre-trained model to target domain distribution at test-time improves model generalization to unseen distribution shift. Widely adopted test-time adaptation (TTA) protocol simultaneously evaluate on a stream of testing data and update model weights (Sun et al. 2020; Wang et al. 2021; Iwasawa and Matsuo 2021; Su, Xu, and Jia 2022; Gandelsman et al. 2022; Goyal et al. 2022; Chen et al. 2022). The state-of-the-art approaches towards TTA adopt self-training (Wang et al. 2021; Su et al. 2023; Gandelsman et al. 2022), distribution alignment (Sun et al. 2020; Su, Xu, and Jia 2022) and self-supervised learning (Liu et al. 2021; Chen et al. 2022). With the above techniques, generalization performance on testing data with corruptions has been substantially improved. Nonetheless, most of these are optimized towards the vanilla TTA protocol, thus these methods may not maintain the superior performance under more realistic TTA scenarios.

Real-World Test-Time Adaptation: Deploying TTA methods in real-world application requires tackling commonly encountered challenges. Recent works summarized multiple challenges that could appear in real-world test-time adaptation, including updating with small batchsize (Niu et al. 2023), non-i.i.d. or temporally correlated testing data (Gong et al. 2022; Wang et al. 2022b; Yuan, Xie, and Li 2023; Boudiaf et al. 2022) and continually adapting to shifting domains (Wang et al. 2022b; Yuan, Xie, and Li 2023; Brahma and Rai 2023). Empirical observations demonstrate that these real-world challenges could pose great challenges to existing TTA methods. Despite the recent efforts in developing TTA robust to non-i.i.d. testing data, we argue that a systematic investigation into more diverse real-world challenges, including global class-imbalance, is missing. This work propose a principled way to simulate these challenges and develop a self-training based method with balanced batchnorm to achieve the state-of-the-art performance.

Methodology

Real-World TTA Protocol

We denote a stream of testing data as $\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_T$ where each $\mathbf{x}_t$ is assumed to be drawn from a distribution $\mathcal{P}(\mathbf{x}|d_t, y_t)$ conditioned on two time-varying variables, namely the testing domain indicator $d_t \in \{1, \cdots K_d\}$ and the class label $y_t \in \{1, \cdots K_c\}$, where K_d and K_c refer to the number of domains (e.g. type of corruptions) and number of semantic classes. In the real-world TTA protocol, both the testing domain indicator and class label distribution could be subject to constant shift, in particular, we assume the domain indicator to exhibit a gradual and slowly shift over time. This is manifested by many real-world applications, e.g. the lighting and weather conditions often changes slowly. We further point out that testing samples are often class imbalanced both locally within a short period of time and globally over the whole testing data stream. Therefore, we model the testing data stream as sampling from a hierarchical probabilistic model. Specifically, we denote a prior $\alpha \in \mathbb{R}^{K_c}$ parameterizing a Dirichlet distribution $\mathbf{q}_c \sim Dir(K_c, \alpha)$. Within a stationary local time window, e.g. a minibatch of testing samples, the labels of testing samples are drawn from a categorical distribution $y \sim Cat(K_c, \mathbf{q}_c)$ where $\mathbf{q}_c$ is drawn from the conjugate prior distribution $Dir(K_c, \alpha)$. The corrupted testing sample is then assumed to be finally sampled from a complex distribution conditioned on the domain indicator d_t and class label y_t, written as $\mathbf{x} \sim \mathcal{P}(\mathbf{x}|d_t, y_t)$. The domain indicator can be modeled as another categorical distribution parameterized by a fixed probability $d_t \sim Cat(K_d, \mathbf{q}_d)$. A

hierarchical probabilistic model simulating the real-world TTA protocol is presented in Fig. 2. We notice the probabilistic model can instantiate multiple existing TTA protocols. For instance, when testing data are locally class imbalanced, as specified by (Yuan, Xie, and Li 2023; Gong et al. 2022), a uniform proportion parameter $\alpha = \sigma\mathbf{1}$ is chosen with a scale parameter σ controlling the degree of local imbalanceness and $\mathbf{q}_c$ is re-sampled every mini-batch. We can easily simulate global class-imbalance by specifying a non-uniform α. We defer a more detailed discussion of simulating real-world TTA protocols with the hierarchical probabilistic model to the supplementary.

Balanced Batch Normalization

Batch normalization (BN) (Ioffe and Szegedy 2015) plays a critical role in enabling more stable model training, reducing sensitivity to hyper-parameters and initialization. When testing data features a distribution shift from the source training data, the regular practice of freezing BN statistics for inference fails to maintain the generalization (Yuan, Xie, and Li 2023; Nado et al. 2020; Gong et al. 2022; Lim et al. 2023; Niu et al. 2023). Hence, adapting BN statistics to testing data distribution becomes a viable solution to TTA (Nado et al. 2020). To adapt model subject to locally imbalanced testing data, the robust batch normalization (Yuan, Xie, and Li 2023) updates BN's mean and variance in a moving average manner on testing data. The robust BN smooths out the bias estimated within each minibatch and achieves competitive performance under non-i.i.d. testing data stream.

Despite being successful in non-i.i.d. testing data stream, a naive moving average update policy struggles in adapting to globally imbalanced testing domain. For example, evidenced in the empirical evaluation in Tab. 1, the performance of RoTTA (Yuan, Xie, and Li 2023) degenerates substantially under more severely global imbalanced testing data. We ascribe the poor performance to the fact that a single BN will bias towards majority classes and normalizing samples from the minority classes with biased statistics will result in severe covariate shift within internal representations. This will eventually cause mis-classifying the minority classes and lower the macro average accuracy. To remedy the bias in adapting BN statistics, we propose a Balanced Batchnorm layer which maintains K_c pairs of statistics separately for each semantic class, denoted as $\{\mu_k\}_{k=1\cdots K_c}$, $\{\sigma_k\}_{k=1\cdots K_c}$. To update category-wise statistics, we apply an efficient iterative updating approach with the help of pseudo labels predictions as follows,

$$\begin{aligned}
&\mu_k^t = \mu_k^{t-1} + \delta_k, \\
&\sigma_k^{2^t} = \sigma_k^{2^{t-1}} - \delta_k^2 + \eta \sum_{b=1}^{B} \mathbb{1}(\hat{y}_b = k) \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \left[(F_{bhw} - \mu_k^{t-1})^2 - \sigma_k^{2^{t-1}} \right] \\
&s.t. \quad \delta_k = \eta \sum_{b=1}^{B} \mathbb{1}(\hat{y}_b = k) \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (F_{bhw} - \mu_k^{t-1}),
\end{aligned} \tag{1}$$

where $\mathbf{F} \in \mathbb{R}^{B\times C\times H\times W}$ denotes the input for Balanced BN layer and $\hat{y}_b$ is the pseudo label predicted by the adapted model in the inference step. With the above design BN statistics for each individual class is separately updated and the global BN statistics are derived from all category-wise statistics as in Eq. 2.

$$\mu_g = \frac{1}{K_c} \sum_{k=1}^{K_c} \mu_k^t, \quad \sigma_g^2 = \frac{1}{K_c} \sum_{k=1}^{K_c} \left[\sigma_k^{2^t} + (\mu_g - \mu_k^t)^2 \right]. \tag{2}$$

Nevertheless, we found when the number of categories is large or the pseudo labels are highly untrustworthy, e.g. the baseline accuracy on ImageNet-C is very low, the above updating strategy might be less effective due to its reliance on the pseudo labels. Therefore, we combine the class-agnostic updating strategy (Robust BN) and the category-wise updating strategy with a balancing parameter γ as below.

$$\mu_k^t = \mu_k^{t-1} + (1-\gamma)\delta_k + \gamma \frac{1}{K_c} \sum_{k'=1}^{K_c} \delta_{k'}, \tag{3}$$

$$\begin{aligned}
&\sigma_k^{2^t} = \sigma_k^{2^{t-1}} + \\
&(1-\gamma) \left\{ -\delta_k^2 + \eta \sum_{b=1}^{B} \mathbb{1}(\hat{y}_b = k) \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \left[(F_{bhw} - \mu_k^{t-1})^2 - \sigma_k^{2^{t-1}} \right] \right\} + \\
&\gamma \cdot \frac{1}{K_c} \sum_{k'=1}^{K_c} \left\{ -\delta_{k'}^2 + \eta \sum_{b=1}^{B} \mathbb{1}(\hat{y}_b = k') \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \left[(F_{bhw} - \mu_{k'}^{t-1})^2 - \sigma_k^{2^{t-1}} \right] \right\}.
\end{aligned} \tag{4}$$

Specifically, when $\gamma = 0$ the updating strategy is the pure class-wise updating strategy and when $\gamma = 1$ the updating strategy degrades to the rule in Robust BN. In all experiments of this paper, we leverage $\gamma = 0.0$ in CIFAR10-C, $\gamma = 0.1$ in CIFAR100-C due to the large number of class and $\gamma = 0.5$ in ImageNet-C due to the highly untrustworthy pseudo labels. The instance-level momentum coefficient η in Balanced BN is set to $0.0005 \times K_c$.

Tri-Net Self-Training

Self-Training (ST) has demonstrated tremendous effectiveness in multiple tasks (Sohn et al. 2020; Kumar, Ma, and Liang 2020). ST updates the model through constraining the prediction consistency between original samples and corresponding augmented samples. In this work, we adopt an approach similar to semi-supervised learning (Sohn et al. 2020) to fine-tune the model to adapt the testing data. In specific, as illustrated in Fig. 3, we introduce teacher $f_t(\mathbf{x};\Theta)$ and student $f_s(\mathbf{x};\Theta)$ networks where the BN layers are independently updated while other weights are shared. The pseudo labels for testing sample are predicted by the teacher network and only the confident pseudo labels are employed for training the student network. Specifically, we denote the probabilistic posterior as $\mathbf{p} = h(f(\mathbf{x}))$ and define the self-training loss in Eq. 5, where $\mathbf{p}^s = h(f_s(\mathcal{A}(\mathbf{x});\Theta))$, $\mathbf{p}^t = h(f_t(\mathbf{x};\Theta))$, $\hat{\mathbf{p}}^t$ refers to the one-hot pseudo label of $\mathbf{p}^t$, $\mathcal{A}$ refers to a strong data augmentation operation, $\mathcal{H}$ refers to entropy and cross-entropy losses and H_0 defines a thresholding hyper-parameter.

$$\mathcal{L}_{st} = \frac{\sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(\mathbf{p}_b^t) < H_0 \cdot \log K_c) \cdot \mathcal{H}(\hat{\mathbf{p}}_b^t, \mathbf{p}_b^s)}{\sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(\mathbf{p}_b^t) < H_0 \cdot \log K_c)} \tag{5}$$

A recent study revealed that self-training is effective for TTA (Su et al. 2023), however without additional regularizations self-training is easily subject to confirmation bias (Arazo et al. 2020). This issue would only exacerbate

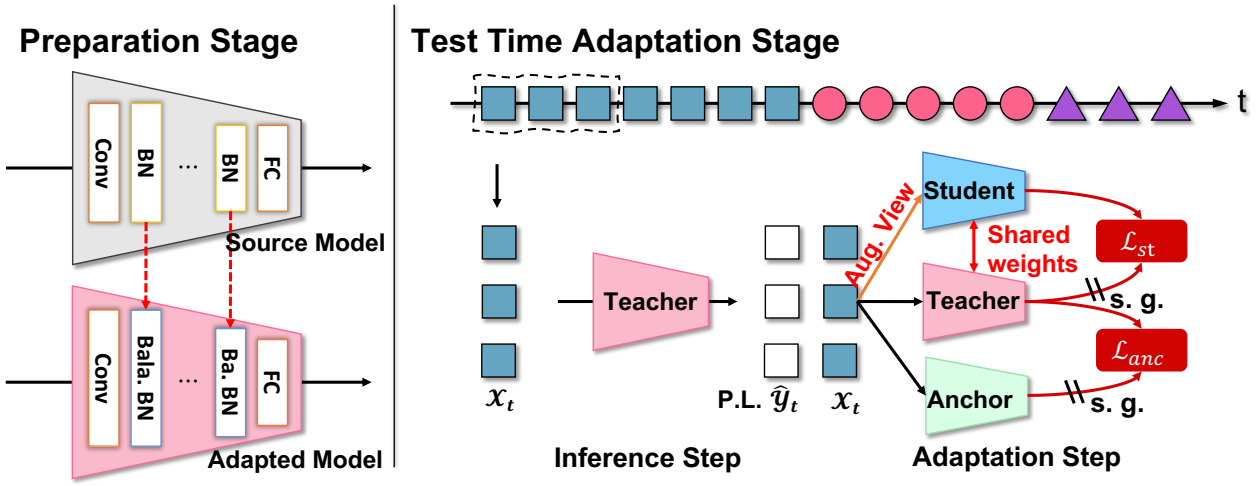

Figure 3: Illustration of the proposed method. We replace the Batchnorm layer of the source model with our proposed Balanced Batchnorm for imbalanced testing set. During test time adaptation, we optimize the combination of self-training loss $\mathcal{L}_{st}$ and anchor loss $\mathcal{L}_{anc}$.

when test data distribution is highly imbalanced, thus leading to over adaptation or collapsed predictions. To avoid over adapting model to a certain test domain, we further propose to incorporate an additional network branch as anchor for regularization.

Anchor Network: We use a frozen source domain network as the anchor network to regularize self-training. In particular, we copy the source model weights, freeze all weights and swap regular BN layers with the proposed Balanced BN layers. To regularize self-training, we design an anchored loss as the mean square error between the posterior predictions of teacher and anchor networks as in Eq. 6. As three network branches are jointly utilized, it gives rise to the term of tri-net self-training.

$$\mathcal{L}_{anc} = \frac{\sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(\mathbf{p}_b^t) < H_0 \cdot \log K_c) ||\mathbf{p}_b^t - \mathbf{p}_b^a||_2^2}{K_c \sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(\mathbf{p}_b^t) < H_0 \cdot \log K_c)} \quad (6)$$

We finally simultaneously optimize the self-training and anchored losses $\mathcal{L} = \mathcal{L}_{st} + \lambda_{anc}\mathcal{L}_{anc}$ w.r.t. the affine parameters of the Balanced BN layers for efficient test-time adaptation.

Experiment

Experiment Settings

Datasets: We evaluate on four test-time adaptation datasets, including **CIFAR10-C** (Hendrycks and Dietterich 2019), **CIFAR100-C** (Hendrycks and Dietterich 2019), **ImageNet-C** (Hendrycks and Dietterich 2019) and **MNIST-C** (Mu and Gilmer 2019). Each of these benchmarks includes 15 types of corruptions with 5 different levels of severity. CIFAR10/100-C both have 10,000 testing samples evenly divided into 10/100 classes for each type of corruptions. ImageNet-C has 5,000 testing samples for each corruption unevenly divided into 1,000 classes [1]. We evaluate all methods under the largest corruption severity level 5 and report the classification error rate (%) throughout the experiment section. We include the detailed results of **MNIST-C** (Mu and Gilmer 2019) in the supplementary.

Hyper-parameters: For CIFAR10-C and CIFAR100-C experiments, we follow the official implementations from previous TTA works (Wang et al. 2021, 2022b; Yuan, Xie, and Li 2023) and respectively adopt a standard pre-trained WideResNet-28 (Zagoruyko and Komodakis 2016) and ResNeXt-29 (Xie et al. 2017) models from RobustBench (Croce et al. 2021) benchmark, for the fair comparison. For ImageNet-C experiments, the standard pre-trained ResNet-50 (He et al. 2016) model in torchvision is adopted. For most competing methods and our TRIBE, we leverage the Adam (Kingma and Ba 2014) optimizer with the learning rate 1e-3 in CIFAR10/100-C and ImageNet-C experiments. As an exception, for Note (Gong et al. 2022) and TTAC (Su, Xu, and Jia 2022) we use the learning rate released in their official implementations. We use a batchsize of 64 for CIFAR10/100-C and 48 for ImageNet-C. Other hyper-parameters of our proposed model are listed as follow: $\lambda_{anc} = 0.5, \eta = 0.0005 \times K_c$ in all datasets, in CIFAR10-C $H_0 = 0.05, \gamma = 0.$, in CIFAR100-C $H_0 = 0.2, \gamma = 0.1$ and in ImageNet-C $H_0 = 0.4, \gamma = 0.5$. Adequate hyper-paramter analysis, provided in the supplementary, demonstrate that the hyper-parameters used into TRIBE are not sensitive. The data augmentations used in TRIBE are described in the supplementary. All of our experiments can be performed on a single NVIDIA GeForce RTX 3090 card.

TTA Evaluation Protocol: We evaluate under two real-world TTA protocols, namely the **GLI-TTA-F** and **GLI-TTA-V**. For both protocols, we create a global class imbalanced testing set following the long-tail dataset creation protocol (Cui

[1] ImageNet-C is only evaluated in a subset with 5,000 testing samples on RobustBench: https://github.com/RobustBench/robustbench

Method	Fixed Global Class Distribution (GLI-TTA-F)			
	$I.F. = 1$	$I.F. = 10$	$I.F. = 100$	$I.F. = 200$
TEST	43.50 / 43.50	42.64 / 43.79	41.71 / 43.63	41.69 / 43.47
BN	75.20 / 75.20	70.77 / 66.77	70.00 / 50.72	70.13 / 47.34
PL	82.90 / 82.90	72.43 / 70.59	70.09 / 55.29	70.38 / 49.86
TENT	86.00 / 86.00	78.15 / 74.90	71.10 / 58.59	69.15 / 53.37
LAME	39.50 / 39.50	38.45 / 40.07	37.48 / 41.80	37.52 / 42.59
COTTA	83.20 / 83.20	73.64 / 71.48	71.32 / 56.44	70.78 / 49.98
NOTE	31.10 / 31.10	36.79 / 30.22	42.59 / 30.75	45.45 / 31.17
TTAC	23.01 / 23.01	31.20 / 29.11	43.40 / 37.37	46.27 / 38.75
PETAL	81.05 / 81.05	73.97 / 71.64	71.14 / 56.11	71.05 / 50.57
RoTTA	25.20 / 25.20	27.41 / 26.31	30.50 / 29.08	32.45 / 30.04
TRIBE	**16.14 / 16.14**	**20.98 / 22.49**	**19.53 / 24.66**	**19.16 / 24.00**

Method	Time-Varying Global Class Distribution (GLI-TTA-V)			
	$I.F. = 1$	$I.F. = 10$	$I.F. = 100$	$I.F. = 200$
TEST	43.50 / 43.50	41.95 / 43.65	40.74 / 43.83	40.53 / 43.77
BN	75.20 / 75.20	71.36 / 67.70	70.35 / 53.07	70.88 / 50.67
PL	82.90 / 82.90	74.74 / 72.12	73.03 / 57.53	72.49 / 54.20
TENT	86.00 / 86.00	77.69 / 74.23	72.99 / 58.65	73.45 / 54.96
LAME	39.50 / 39.50	38.02 / 40.15	36.51 / 42.16	36.24 / 42.16
COTTA	83.20 / 83.20	75.29 / 71.87	73.83 / 56.80	74.97 / 56.47
NOTE	31.10 / 31.10	29.52 / 29.23	30.02 / 29.88	29.71 / 30.28
TTAC	23.01 / 23.01	32.25 / 32.12	36.84 / 37.13	37.96 / 38.07
PETAL	81.05 / 81.05	75.19 / 71.65	72.71 / 55.73	73.76 / 53.51
RoTTA	25.20 / 25.20	27.61 / 26.35	32.16 / 29.32	33.34 / 31.35
TRIBE	**16.14 / 16.14**	**20.92 / 22.40**	**22.44 / 25.50**	**23.10 / 27.03**

Table 1: Average classification error on CIFAR10-C while continually adapting to different corruptions at the highest severity 5 with globally and locally class-imbalanced test stream. $I.F.$ is the Imbalance Factor of Global Class Imbalance. Instance-wise average error rate $a\%$ and category-wise average error rate $b\%$ are separated by (a / b).

Method	Fixed Global Class Distribution (GLI-TTA-F)			
	$I.F. = 1$	$I.F. = 10$	$I.F. = 100$	$I.F. = 200$
TEST	46.40 / 46.40	46.96 / 46.52	47.53 / 45.91	47.59 / 39.94
BN	52.90 / 52.90	46.05 / 42.29	47.01 / 40.01	47.38 / 35.26
PL	88.90 / 88.90	68.51 / 69.71	53.46 / 57.26	49.41 / 49.26
TENT	92.80 / 92.80	76.88 / 79.08	56.72 / 65.96	50.45 / 58.45
LAME	40.50 / 40.50	43.66 / 44.88	44.15 / 46.64	43.81 / 40.33
COTTA	52.20 / 52.20	44.48 / 40.93	45.46 / 38.77	45.67 / 33.72
NOTE	73.80 / 73.80	57.71 / 58.86	54.44 / 57.10	53.74 / 52.48
TTAC	34.10 / 34.10	40.48 / 38.28	47.84 / 41.47	49.78 / 38.00
PETAL	55.03 / 55.03	45.14 / 41.91	44.63 / 38.52	44.75 / 33.81
RoTTA	35.00 / 35.00	40.00 / 39.03	45.68 / 42.04	46.78 / 37.93
TRIBE	**33.26 / 33.26**	**33.10 / 34.31**	**32.31 / 34.98**	**32.29 / 31.54**

Method	Time-Varying Global Class Distribution (GLI-TTA-V)			
	$I.F. = 1$	$I.F. = 10$	$I.F. = 100$	$I.F. = 200$
TEST	46.40 / 46.40	45.85 / 46.65	45.34 / 46.94	45.16 / 40.61
BN	52.90 / 52.90	45.10 / 42.47	45.15 / 38.80	45.37 / 33.45
PL	88.90 / 88.90	68.16 / 66.62	52.83 / 48.39	53.68 / 44.28
TENT	92.80 / 92.80	77.11 / 76.51	65.42 / 63.48	62.45 / 53.57
LAME	40.50 / 40.50	42.82 / 45.35	42.47 / 47.82	42.23 / 41.45
COTTA	52.20 / 52.20	43.74 / 41.03	43.83 / 37.93	43.96 / 32.69
NOTE	73.80 / 73.80	58.07 / 58.46	55.16 / 55.95	54.43 / 48.65
TTAC	34.10 / 34.10	38.56 / 38.68	42.07 / 41.05	42.87 / 35.80
PETAL	55.03 / 55.03	44.36 / 41.54	44.11 / 38.33	44.43 / 32.84
RoTTA	35.00 / 35.00	39.56 / 39.77	42.20 / 39.93	43.57 / 35.82
TRIBE	**33.26 / 33.26**	**33.52 / 34.49**	**33.76 / 34.63**	**34.29 / 30.22**

Table 2: Average classification error on CIFAR100-C while continually adapting to different corruptions at the highest severity 5 with globally and locally class-imbalanced test stream.

et al. 2019), we choose three imbalance factor $I.F.$ as 1, 10, 100 and 200 for evaluation where GLI-TTA degrades into PTTA setting (Yuan, Xie, and Li 2023) with $I.F. = 1$. A default scale parameter $\sigma = 0.1$ is chosen to control local class imbalance. To simulate continually shifting domains, we sample without replacement the domain indicator after all testing samples are predicted. For better reproducibility we provide the sequence of domains in the supplementary. Under the GLI-TTA-F setting, we fix the proportion parameter α throughout the experiment. Under the GLI-TTA-V setting, we randomly permute class indices after adaptation to each domain (type of corruption) to simulate time-varying class distribution.

Competing Methods: We benchmark against the following TTA methods (Nado et al. 2020; Lee et al. 2013; Wang et al. 2021; Boudiaf et al. 2022; Gong et al. 2022; Su, Xu, and Jia 2022; Yuan, Xie, and Li 2023; Niu et al. 2022). Direct testing (**TEST**) performs inference on test streaming data without adaptation. Prediction-time batch normalization (**BN**) (Nado et al. 2020) replaces the running statistics with the batch statistics on each testing minibatch for normalization. Pseudo Label (**PL**) (Lee et al. 2013) updates the parameters of all normalization layers by minimizing the cross-entropy loss with predicted pseudo labels. Test-time entropy minimization (**TENT**) (Wang et al. 2021) updates the affine parameters of all batchnorm layers by minimizing the entropy of predictions. Laplacian adjusted maximum-likelihood estimation (**LAME**) (Boudiaf et al. 2022) adjusts the predictions of the model through maximizing the likelihood estimation without updating any parameters. Continual test-time adaptation (**CoTTA**) (Wang et al. 2022b) performs mean-teacher architecture, and randomly selects and restores the parameters of the model to source model. **PETAL** (Brahma and Rai 2023) leverages fisher information to instruct the parameter restoration. Non-i.i.d. test-time adaptation (**NOTE**) (Gong et al. 2022) optionally updates the batchnorm statistics when the distance between the instance statistics of the test sample and the source model's statistics is less than a threshold. Test-time anchored clustering (**TTAC**) (Su, Xu, and Jia 2022) minimizes the KL-Divergence between the source and target domain distributions. Robust test-time adaptation (**RoTTA**) (Yuan, Xie, and Li 2023) replaces the batchnorm layers with Robust Batch Normalization for better estimation of target domain batchnorm statistics. Finally, we evaluate our **TRIBE** with tri-net self-training and Balanced Batchnorm layers.

Real-World Test Time Adaptation Results

Under the proposed real-world test-time adaptation protocol, the classification errors averaged over continuously adapting to all 15 types of corruptions under different degrees of global imbalance are calculated. We report the results in Tab. 1 for CIFAR10-C and Tab. 2 for CIFAR100-C. We make the following observations from the results. i) Direct testing without any adaptation is even stronger than many TTA methods. For example, only LAME, TTAC, RoTTA and our TRIBE

Time	$t \longrightarrow$															
Method	motion	snow	fog	shot	defocus	contrast	zoom	brightness	frost	elastic	glass	gaussian	pixelate	jpeg	impulse	Avg.
TEST	85.15	83.45	75.88	97.09	81.68	94.52	77.93	41.23	77.07	82.48	89.73	97.81	79.31	68.50	98.17	82.00
BN	73.64	66.07	52.81	84.49	85.05	82.66	61.96	36.04	68.60	56.44	84.85	85.31	52.29	60.77	85.05	69.07
PL	66.55	60.43	49.46	**76.57**	79.23	81.04	65.35	51.48	75.62	69.74	89.04	92.36	86.84	92.09	97.83	75.58
TENT	**64.37**	59.73	51.20	77.47	81.70	88.72	82.38	76.91	93.64	95.43	98.80	98.98	98.39	98.90	99.40	84.40
LAME	85.93	84.57	77.29	97.47	81.92	94.72	78.41	41.49	77.67	84.07	90.25	98.21	79.61	68.64	98.76	82.60
EATA	73.15	65.41	52.51	84.27	85.09	82.85	61.52	35.15	68.26	56.30	84.43	84.95	51.63	60.85	85.05	68.76
NOTE	82.97	78.29	73.43	93.92	96.35	89.73	93.18	84.57	92.82	94.54	98.50	98.88	98.17	97.55	98.78	91.44
ROTTA	74.86	70.02	55.26	85.55	85.37	78.61	61.00	**34.31**	64.65	52.83	76.16	85.43	**48.70**	52.41	78.37	66.90
TRIBE	69.52	**59.55**	**48.35**	79.27	**78.47**	**75.54**	**56.62**	35.19	**60.39**	**49.26**	**74.54**	**74.10**	50.08	**51.24**	**72.59**	**62.32**

Table 3: Average classification error on ImageNet-C while continually adapting to different corruptions at the highest severity 5 with non-i.i.d. testing data stream.

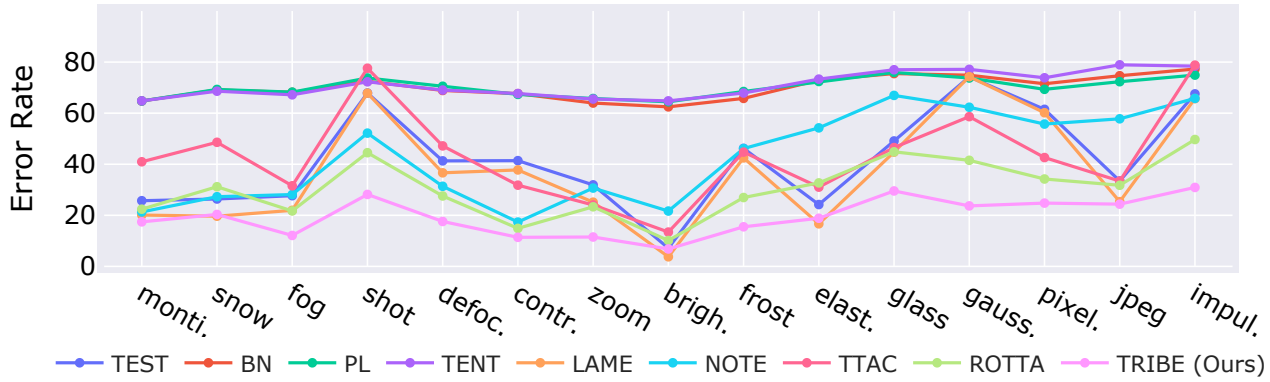

Figure 4: Performances on each individual domain (corruption) under GLI-TTA-F (I.F.=100) protocols on CIFAR10-C dataset.

could consistently outperform direct testing (TEST) on both CIFAR10-C and CIFAR100-C datasets, suggesting the necessity to develop robust TTA approaches. ii) Global class imbalance poses a great challenge to existing robust TTA methods. For example, the previous state-of-the-art, RoTTA achieves 25.2% and 35.0% on CIFAR10-C and CIFAR100-C respectively, while the error rose to 30.04% and 37.93% under severely global imbalanced testing set ($I.F. = 200$). The same observation applies to other competing methods. In comparison, TRIBE is able to maintain relatively better performance under more severe global imbalanced testing set. iii) We further notice that TRIBE consistently outperform all competing methods in absolute accuracy. Importantly, under balanced global distribution ($I.F. = 1$), TRIBE outperforms the best performing model, TTAC, by 7% on CIFAR10-C. The margin is maintained under more imbalanced testing set ($I.F. = 200$). iv) TRIBE maintains a more consistent performance from $I.F. = 10$ to $I.F. = 200$ on both CIFAR10-C and CIFAR100-C, while other competing methods degenerate substantially. This is attributed to the introduction of Balanced BN layer better accounting for severe class imbalance and anchored loss avoiding over adaptation across the different domains.

We further evaluate TTA performance on ImageNet-C dataset of which the testing set is naturally class imbalanced. Therefore, we only simulate local class imbalance for the testing data stream and allow α equal to the marginalized class distribution. We present both averaged and domain specific classification error in Tab. 3. We make similar observations with results on CIFAR10/100-C. Some competitive TTA methods perform exceptionally worse than direct testing while TRIBE again outperforms all competing methods both in terms of averaged error rate and winning on 11/15 corruption types.

Results on Individual Corruption: We adapt the model continually to constant shifting domains (corruption types). We report the average classification error for each individual type of corruptions in Fig. 4. We conclude from the plots that i) BN, PL and TENT normalize the features using the statistics calculated within current mini-batch, thus they all perform much worse than methods considering robust batchnorm e.g. NOTE, ROTTA and TRIBE. ii) There is a strong correlation of performance across different methods suggesting certain corruptions, e.g. "shot", "gaussian noise" and "impulse noise", are inherently more difficult. Nevertheless, TRIBE always outperforms competing methods on these challenging corruptions. iii) Some competing methods achieve close to TRIBE accuracy on easier corruptions, but they often perform much worse on the upcoming corruptions. Overall, TRIBE exhibits much lower variance across all domains when continually adapted. This suggests the anchored loss potentially helps TRIBE to avoid over adapting to easier domains.

Method	EMA Model	BatchNorm	Self-Training	Anchored Loss	CIFAR10-C	CIFAR100-C	Avg.
TEST	–	BN	–	–	41.71 / 43.63	47.53 / 45.91	44.62 / 44.77
ROTTA	✓	Robust BN	✓	–	30.50 / 29.08	45.68 / 42.04	38.09 / 35.56
–	–	Robust BN	–	–	43.48 / 32.29	40.45 / 36.94	41.97 / 34.62
–	–	Balanced BN	–	–	29.00 / 26.38	39.55 / 36.59	34.28 / 31.49
–	–	BN	✓	–	37.67 / 38.94	37.12 / 44.77	37.40 / 41.86
–	–	Balanced BN	✓	–	36.58 / 65.88	37.21 / 44.83	36.90 / 55.36
–	–	BN	✓	✓	36.76 / 29.19	36.16 / 36.26	36.46 / 32.73
MT*	✓	Balanced BN	✓	–	23.76 / 25.18	36.01 / 35.72	29.89 / 30.45
TRIBE	–	Balanced BN	✓	✓	**19.53 / 24.66**	**32.31 / 34.98**	**25.92 / 29.82**

Table 4: Ablation study on CIFAR10/100-C under GLI-TTA-F ($I.F. = 100$) protocol. We report classification error as evaluation metric. MT* indicates Mean Teacher is adapted to TTA task by removing the labeled loss term.

Method	Imbalance Factor			
	1	10	100	200
TEST	43.51	42.65	41.71	41.69
IABN (NIPS22)	27.29	31.74	38.19	40.03
Robust BN (5e-2)	46.33	39.49	43.48	45.39
Robust BN (5e-3)	29.50	36.38	44.36	46.71
Balanced BN (Ours)	**24.71**	**29.96**	**29.00**	**30.24**

Table 5: The performance of different normalization layers which only updates the statistics. The classification error on CIFAR10-C are reported.

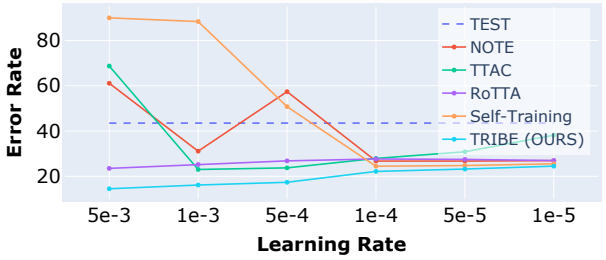

Figure 5: We evaluate state-of-the-art TTA methods under different learning rates. The learning rates of NOTE fall in [5e-4, 1e-6] and TTAC fall in [5e-5, 1e-7] in order to align the best LR with other methods.

Ablation & Additional Study

Effect of Individual Components: We investigate the effectiveness of proposed components in Tab. 4. Specifically, we first compare adaptation by updating batchnorm statistics. It is apparent that Balanced BN is substantially better than Robust BN (Yuan, Xie, and Li 2023) when separately applied. When a two branch self-training (teacher & student net) is applied, we witness a clear improvement from the direct testing baseline. However the improvement is less significant by combining self-training with Balanced BN. This is probably caused by over adaptation to testing domains causing poor generalization to continually changing domains. This negative impact is finally remedied by introducing a tri-net architecture (Anchored Loss) which helps regularize self-training to avoid over adaptation.

Comparing Batchnorm Layers: To evaluate the effectiveness of our proposed Balanced BN, we run forward pass for global and local class-imbalanced testing samples for multiple batch normalization modules proposed for real-world TTA, with results presented in Tab. 5. We observe our proposed Balanced BN outperforms others with a large margin ($2.58 \sim 9.79\%$), especially under severely global class imbalance ($I.F. = 200$). It further confirms that Balanced BN is more suitable for handling both global and local class-imbalanced testing data.

Hyper-parameter Robustness: Selecting appropriate hyper-parameter plays an important role in TTA (Zhao et al. 2023). As TTA assumes no labeled data in testing set, selecting appropriate hyper-parameter becomes non-trivial. We argue that the tri-net design is naturally more robust to the choice of learning rate. As illustrated in Fig. 5, TRIBE is very stable w.r.t. the choice of learning rate while other methods, e.g. TTAC and NOTE, prefer a much narrower range of learning rate. More hyper-parameter analysis details can be found in the supplementary.

Conclusion

In this work, we explore improving test-time adaptation algorithm's robustness to real-world challenges, including non-i.i.d. testing data stream, global class imbalance and continual domain shift. To adapt to imbalanced testing data, we propose a Balanced Batchnorm layer consisting of multiple category-wise statistics to achieve unbiased estimation of statistics. We further propose a tri-net architecture with student, teacher and anchor networks to regularize self-training based TTA. We demonstrate the effectiveness of the overall method, TRIBE, on simulated real-world test-time adaptation data streams. We achieve the state-of-the-art performance on all benchmarks created from four TTA datasets.

Limitations: TRIBE replaces regular Batchnorm layer with a customized Balanced Batchnorm layer, thus introducing additional storage overhead. Moreover, some recent Transformer based backbone network prefer Layernorm to Batchnorm (Dosovitskiy et al. 2021), thus potentially limiting the application of TRIBE. But recent studies revealed opportunities to integrate batchnorm to vision Transformer networks (Yao et al. 2021).

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) (Grant Number: 62106078), and Agency for Science, Technology and Research (Grant Number: C210112059). This work was partially done during Yongyi Su's attachment with Institute for Infocomm Research (I2R), funded by China Scholarship Council (CSC).

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks*.

Boudiaf, M.; Mueller, R.; Ben Ayed, I.; and Bertinetto, L. 2022. Parameter-free Online Test-time Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Brahma, D.; and Rai, P. 2023. A Probabilistic Framework for Lifelong Test-Time Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3582–3591.

Chen, D.; Wang, D.; Darrell, T.; and Ebrahimi, S. 2022. Contrastive Test-Time Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2021. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Gandelsman, Y.; Sun, Y.; Chen, X.; and Efros, A. A. 2022. Test-time training with masked autoencoders. In *Advances in Neural Information Processing Systems*.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*.

Gong, T.; Jeong, J.; Kim, T.; Kim, Y.; Shin, J.; and Lee, S. 2022. Robust Continual Test-time Adaptation: Instance-aware BN and Prediction-balanced Memory. In *Advances in Neural Information Processing Systems*.

Goyal, S.; Sun, M.; Raghunathan, A.; and Kolter, J. Z. 2022. Test Time Adaptation via Conjugate Pseudo-labels. In *Advances in Neural Information Processing Systems*.

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

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

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hoffman, J.; Tzeng, E.; Park, T.; Zhu, J.-Y.; Isola, P.; Saenko, K.; Efros, A.; and Darrell, T. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*.

Iwasawa, Y.; and Matsuo, Y. 2021. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34: 2427–2440.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kumar, A.; Ma, T.; and Liang, P. 2020. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning*.

Lee, D.-H.; et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 896.

Lim, H.; Kim, B.; Choo, J.; and Choi, S. 2023. TTN: A domain-shift aware batch normalization in test-time adaptation. *arXiv preprint arXiv:2302.05155*.

Liu, H.; Wang, J.; and Long, M. 2021. Cycle self-training for domain adaptation. In *Advances in Neural Information Processing Systems*.

Liu, Y.; Kothari, P.; van Delft, B.; Bellot-Gurlet, B.; Mordan, T.; and Alahi, A. 2021. TTT++: When Does Self-Supervised Test-Time Training Fail or Thrive? In *Advances in Neural Information Processing Systems*.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*.

Mu, N.; and Gilmer, J. 2019. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*.

Nado, Z.; Padhy, S.; Sculley, D.; D'Amour, A.; Lakshminarayanan, B.; and Snoek, J. 2020. Evaluating Prediction-Time Batch Normalization for Robustness under Covariate Shift. *CoRR*, abs/2006.10963.

Niu, S.; Wu, J.; Zhang, Y.; Chen, Y.; Zheng, S.; Zhao, P.; and Tan, M. 2022. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*.

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

Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*.

Su, Y.; Xu, X.; and Jia, K. 2022. Revisiting Realistic Test-Time Training: Sequential Inference and Adaptation by Anchored Clustering. In *Advances in Neural Information Processing Systems*.

Su, Y.; Xu, X.; Li, T.; and Jia, K. 2023. Revisiting Realistic Test-Time Training: Sequential Inference and Adaptation by Anchored Clustering Regularized Self-Training. *arXiv preprint arXiv:2303.10856*.

Sun, B.; and Saenko, K. 2016. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*.

Sun, J.; Zhang, Q.; Kailkhura, B.; Yu, Z.; Xiao, C.; and Mao, Z. M. 2022. Benchmarking Robustness of 3D Point Cloud Recognition Against Common Corruptions. *arXiv preprint arXiv:2201.12296*.

Sun, Y.; Wang, X.; Liu, Z.; Miller, J.; Efros, A.; and Hardt, M. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*.

Tang, H.; Chen, K.; and Jia, K. 2020. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; and Darrell, T. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Wang, D.; Shelhamer, E.; Liu, S.; Olshausen, B. A.; and Darrell, T. 2021. Tent: Fully Test-Time Adaptation by Entropy Minimization. In *International Conference on Learning Representations*.

Wang, J.; Lan, C.; Liu, C.; Ouyang, Y.; Qin, T.; Lu, W.; Chen, Y.; Zeng, W.; and Yu, P. 2022a. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, M.; and Deng, W. 2018. Deep visual domain adaptation: A survey. *Neurocomputing*.

Wang, Q.; Fink, O.; Gool, L. V.; and Dai, D. 2022b. Continual Test-Time Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1492–1500.

Yao, Z.; Cao, Y.; Lin, Y.; Liu, Z.; Zhang, Z.; and Hu, H. 2021. Leveraging batch normalization for vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*.

Yuan, L.; Xie, B.; and Li, S. 2023. Robust Test-Time Adaptation in Dynamic Scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *British Machine Vision Conference 2016*. British Machine Vision Association.

Zellinger, W.; Grubinger, T.; Lughofer, E.; Natschläger, T.; and Saminger-Platz, S. 2016. Central moment discrepancy (cmd) for domain-invariant representation learning. In *International Conference on Learning Representations*.

Zhao, H.; Liu, Y.; Alahi, A.; and Lin, T. 2023. On Pitfalls of Test-Time Adaptation. In *ICLR 2023 Workshop on Pitfalls of limited data and computation for Trustworthy ML*.