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

Yongyi Su1, Xun Xu2, 1*, Kui Jia3*

1South China University of Technology
2Institute for Infocomm Research, A*STAR
3School 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,
we adopt a teacher-student framework for TTA. Nonetheless, these approaches still do not explicitly address the challenge of constant shifting domains in 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 Balanced 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...
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 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 proposes 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 $x_0, x_1, \ldots, x_T$ where each $x_t$ is assumed to be drawn from a distribution $P(x|d_t, y_t)$ 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 change slowly. We further point out that testing samples are often class balanced 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 $q_c \sim Dir(K_c, \alpha)$. Within a stationary local time window, e.g., a minibatch of testing samples, the labels of testing samples are drawn from a categorical distribution $y \sim Cat(K_c, q_c)$ where $q_c$ 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 $x \sim P(x|d_t, y_t)$. The domain indicator can be modeled as another categorical distribution parameterized by a fixed probability $d_t \sim Cat(K_d, q_d)$. A
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 $\{\mu_k\}_{k=1,\ldots,K_c}$ and $\{\sigma_k\}_{k=1,\ldots,K_c}$. To update category-wise statistics, we apply an efficient iterative updating approach with the help of pseudo labels predictions as follows,

$$\mu_k^t = \mu_k^{t-1} + \delta_k,$$

$$\sigma_k^t = \sigma_k^{t-1} + \delta_k + \eta \sum_{h=1}^{n} t(y_h = k) \frac{1}{B} \sum_{b=1}^{B} \sum_{w=1}^{W} \left(F_{bhw} - \mu_k^{t-1}\right)^2 - \sigma_k^{t-1},$$

subject to $\delta_k = \eta \sum_{h=1}^{n} t(y_h = k) \frac{1}{B} \sum_{b=1}^{B} \sum_{w=1}^{W} \left(F_{bhw} - \mu_k^{t-1}\right)$. (1)

where $F \in \mathbb{R}^{B \times C \times H \times W}$ denotes the input for Balanced BN layer and $y_h$ 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.

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

Nevertheless, we found when the number of categories is large 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 $\gamma$ as below.

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

$$\sigma_k^t = \sigma_k^{t-1} + \gamma \frac{1}{K_c} \sum_{k'=1}^{K_c} \left(1 - \gamma\right) \delta_{k'},$$

subject to $\delta_k = \gamma \sum_{h=1}^{n} t(y_h = k) \frac{1}{B} \sum_{b=1}^{B} \sum_{w=1}^{W} \left(F_{bhw} - \mu_k^{t-1}\right)^2 - \sigma_k^{t-1}$;

$$\delta_k' = \gamma \sum_{h=1}^{n} t(y_h = k') \frac{1}{B} \sum_{b=1}^{B} \sum_{w=1}^{W} \left(F_{bhw} - \mu_k^{t-1}\right)^2 - \sigma_k^{t-1}.$$ (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 $\eta$ 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 $f_s(x; \Theta)$ and student $f_s(x; \Theta)$ 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 $p = h(f(x))$ and define the self-training loss in Eq. 5, where $p^t = h(f_s(A(x); \Theta))$, $p^t = h(f_s(x; \Theta))$, $p^t$ refers to the one-hot pseudo label of $p^t$, $A$ refers to a strong data augmentation operation, $\mathcal{H}$ refers to entropy and cross-entropy losses and $H_0$ defines a thresholding hyper-parameter.

$$\mathcal{L}_{st} = \frac{\sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(p^t_b) < H_0 \cdot \log K_c) \cdot \mathcal{H}(p^t_b, p_b^t)}{\sum_{b=1}^{B} \mathbb{1}(\mathcal{H}(p^t_b) < H_0 \cdot \log K_c)}.$$ (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
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 regularization.

\[
\mathcal{L}_{\text{anc}} = \frac{1}{K_c} \sum_{b=1}^{B} \left( \mathcal{H}(\mathbf{p}_b^t) < H_0 \cdot \log K_c \right) \left| \mathbf{p}_b^t - \mathbf{p}_b^c \right|^2
\]

We finally simultaneously optimize the self-training and anchored losses \( \mathcal{L} = \mathcal{L}_{st} + \lambda_{\text{anc}} \mathcal{L}_{\text{anc}} \) 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. 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 batch size of 64 for CIFAR10/100-C and 48 for ImageNet-C. Other hyper-parameters of our proposed model are listed as follow: \( \lambda_{\text{anc}} = 0.5, \eta = 0.0005 \times K_c \) in all datasets, in CIFAR10-C \( H_0 = 0.05, \gamma = 0.4 \), in CIFAR100-C \( H_0 = 0.2, \gamma = 0.1 \) and in ImageNet-C \( H_0 = 0.4, \gamma = 0.5 \). Adequate hyper-parameter 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 et al. 2023).
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 \( \alpha \%) and category-wise average error rate \( b\% \) are separated by (a/b).

<table>
<thead>
<tr>
<th>Method</th>
<th>Fixed Global Class Distribution (GLI-TTA-F)</th>
<th>( I.F. = 1 )</th>
<th>( I.F. = 10 )</th>
<th>( I.F. = 100 )</th>
<th>( I.F. = 200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>43.50 / 43.50</td>
<td>42.64 / 43.79</td>
<td>41.71 / 43.63</td>
<td>41.69 / 43.47</td>
<td></td>
</tr>
<tr>
<td>BN</td>
<td>75.20 / 75.20</td>
<td>70.77 / 67.67</td>
<td>70.00 / 50.72</td>
<td>70.13 / 47.34</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>82.90 / 82.90</td>
<td>72.43 / 70.59</td>
<td>70.09 / 55.29</td>
<td>70.38 / 49.86</td>
<td></td>
</tr>
<tr>
<td>TENT</td>
<td>86.00 / 86.00</td>
<td>78.15 / 74.90</td>
<td>71.10 / 58.59</td>
<td>69.15 / 53.37</td>
<td></td>
</tr>
<tr>
<td>LAME</td>
<td>39.30 / 39.30</td>
<td>38.45 / 40.07</td>
<td>37.48 / 41.80</td>
<td>37.52 / 42.59</td>
<td></td>
</tr>
<tr>
<td>COTTA</td>
<td>83.20 / 83.20</td>
<td>73.64 / 71.48</td>
<td>71.32 / 56.44</td>
<td>70.78 / 49.98</td>
<td></td>
</tr>
<tr>
<td>NOTE</td>
<td>31.10 / 31.10</td>
<td>36.79 / 30.22</td>
<td>42.59 / 30.75</td>
<td>45.45 / 31.17</td>
<td></td>
</tr>
<tr>
<td>TTAC</td>
<td>23.01 / 23.01</td>
<td>31.20 / 29.11</td>
<td>43.40 / 37.37</td>
<td>46.27 / 38.75</td>
<td></td>
</tr>
<tr>
<td>PETAL</td>
<td>81.05 / 81.05</td>
<td>73.97 / 71.64</td>
<td>71.14 / 56.11</td>
<td>71.05 / 50.57</td>
<td></td>
</tr>
<tr>
<td>RoTTA</td>
<td>25.20 / 25.20</td>
<td>27.41 / 26.31</td>
<td>30.50 / 29.08</td>
<td>32.45 / 30.04</td>
<td></td>
</tr>
</tbody>
</table>


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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fixed Global Class Distribution (GLI-TTA-F)</th>
<th>( I.F. = 1 )</th>
<th>( I.F. = 10 )</th>
<th>( I.F. = 100 )</th>
<th>( I.F. = 200 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>46.40 / 46.40</td>
<td>46.96 / 46.52</td>
<td>47.53 / 45.91</td>
<td>47.59 / 39.94</td>
<td></td>
</tr>
<tr>
<td>BN</td>
<td>52.90 / 52.90</td>
<td>46.05 / 42.99</td>
<td>47.01 / 40.01</td>
<td>47.38 / 35.26</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>88.90 / 88.90</td>
<td>68.51 / 69.71</td>
<td>53.46 / 57.26</td>
<td>49.41 / 49.26</td>
<td></td>
</tr>
<tr>
<td>TENT</td>
<td>92.80 / 92.80</td>
<td>76.88 / 79.08</td>
<td>56.72 / 65.96</td>
<td>50.45 / 58.45</td>
<td></td>
</tr>
<tr>
<td>LAME</td>
<td>40.50 / 40.50</td>
<td>43.66 / 44.88</td>
<td>44.15 / 46.64</td>
<td>43.81 / 40.33</td>
<td></td>
</tr>
<tr>
<td>COTTA</td>
<td>52.20 / 52.20</td>
<td>44.48 / 40.93</td>
<td>45.46 / 38.77</td>
<td>45.67 / 33.72</td>
<td></td>
</tr>
<tr>
<td>NOTE</td>
<td>73.80 / 73.80</td>
<td>57.71 / 58.86</td>
<td>54.44 / 57.10</td>
<td>53.74 / 52.48</td>
<td></td>
</tr>
<tr>
<td>TTAC</td>
<td>34.10 / 34.10</td>
<td>40.48 / 38.28</td>
<td>47.84 / 41.47</td>
<td>49.78 / 38.00</td>
<td></td>
</tr>
<tr>
<td>PETAL</td>
<td>55.03 / 55.03</td>
<td>45.14 / 41.91</td>
<td>44.63 / 38.52</td>
<td>44.75 / 33.81</td>
<td></td>
</tr>
<tr>
<td>RoTTA</td>
<td>35.00 / 35.00</td>
<td>40.00 / 39.03</td>
<td>45.68 / 42.04</td>
<td>46.78 / 37.93</td>
<td></td>
</tr>
</tbody>
</table>

| TRIBE  | 33.26 / 33.26 | 33.10 / 34.31 | 32.31 / 34.98 | 32.29 / 31.54 |               |

For example, only LAME, TTAC, RoTTA and our TRIBE (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

---

**NOTE**

COTTA (Wang et al. 2022a) replaces the batchnorm layer by performing batch normalization on each mini-batch 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
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 $\alpha$ 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.

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

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

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 ~ 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


