Unraveling Batch Normalization for Realistic Test-Time Adaptation

Zixian Su[1, 2], Jingwei Guo[1, 2], Kai Yao[1, 2], Xi Yang[1*], Qiufeng Wang[1], Kaizhu Huang[3*]

[1]Department of Intelligent Science, Xi'an Jiaotong-Liverpool University, Suzhou, China
[2]Faculty of Science and Engineering, University of Liverpool, Liverpool, the United Kingdom
[3]Data Science Research Center, Duke Kunshan University, Kunshan, China
zixian.su@liverpool.ac.uk, jingwei.guo@liverpool.ac.uk
xi.yang01@xjtlu.edu.cn, kaizhu.huang@dukekunshan.edu.cn

Abstract

While recent test-time adaptations exhibit efficacy by adjusting batch normalization to narrow domain disparities, their effectiveness diminishes with realistic mini-batches due to inaccurate target estimation. As previous attempts merely introduce source statistics to mitigate this issue, the fundamental problem of inaccurate target estimation still persists, leaving the intrinsic test-time domain shifts unresolved. This paper delves into the problem of mini-batch degradation. By unraveling batch normalization, we discover that the inexact target statistics largely stem from the substantially reduced class diversity in batch. Drawing upon this insight, we introduce a straightforward tool, Test-time Exponential Moving Average (TEMA), to bridge the class diversity gap between training and testing batches. Importantly, our TEMA adaptively extends the scope of typical methods beyond the current batch to incorporate a diverse set of class information, which in turn boosts an accurate target estimation. Built upon this foundation, we further design a novel layer-wise rectification strategy to consistently promote test-time performance. Our proposed method enjoys a unique advantage as it requires neither training nor tuning parameters, offering a truly hassle-free solution. It significantly enhances model robustness against shifted domains and maintains resilience in diverse real-world scenarios with various batch sizes, achieving state-of-the-art performance on several major benchmarks. Code is available at https://github.com/kiwi12138/RealisticTTA.

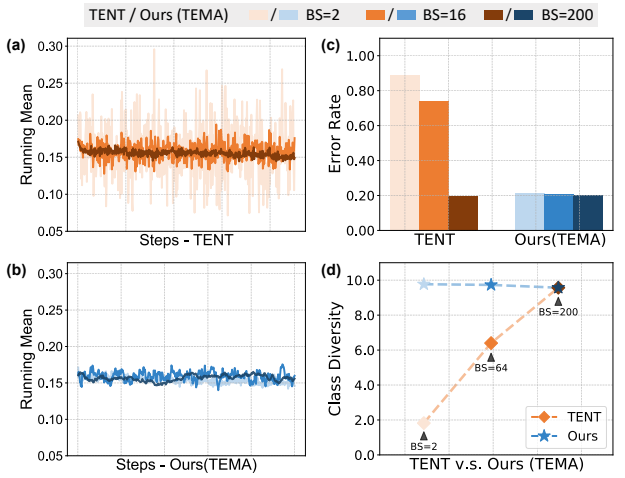


Figure 1: (a)(b) Running mean statistics of one specific channel in the last BN layer during inference. (c) Performance comparison between TENT (Wang et al. 2021) and Ours under different batch sizes. (d) Quantified class diversity of TENT v.s. Ours under different batch sizes. Same color denotes same evaluation setting. As can be seen, class diversity plays a vital role in test-time performance.

Introduction

Confronted with unseen environments, the effectiveness of deep neural networks (Krizhevsky, Sutskever, and Hinton 2017; He et al. 2016; Chen et al. 2017) often suffers a decline due to domain shift (Ganin and Lempitsky 2015; Long et al. 2013) — an incongruity between the training (source) and testing (target) domains. To address this, Test-Time Adaptation (TTA) (Wang et al. 2021; Sun et al. 2020) serves as a practical paradigm that enables pre-trained models to dynamically adapt with test streams. Recent TTA research mainly resolves around exploring batch normalization (BN) (Ioffe and Szegedy 2015). A crucial reason for this focus lies in the internal relation between normalization statistics and domain characteristics. Specifically, BN uses source statistics for normalization during training; however, applying the same parameters to differing target domains can lead to erratic results. Instead, normalizing with the estimated statistics from the target domain promotes appropriate standardization and superior generalization across diverse environments.

Despite their excellent performance under chosen settings, existing TTA methods face essential challenges when deployed in realistic settings (Wang et al. 2022; Lim et al. 2023). Most of them necessitate extensive target data, commonly large batches. While in practice, mini-batches are often preferred, which may produce erroneous statistics and cause model degradation (see Figure 1 (c)). Although some attempts (Schneider et al. 2020; You, Li, and Zhao 2021) have been made to alleviate this limitation by incorporating source statistics, they merely cover up the inaccurate target information, leaving the persistent domain shifts unresolved. These observations naturally prompt us to ask: *What exactly*

*Corresponding author.

causes the inaccurate target statistics? How can we mitigate these inaccuracies to alleviate domain shift?

This paper explores these concerns by unraveling batch normalization. A common belief in TTA research is that the accuracy of target estimation largely hinges on the quantity of batch samples (Zhao, Chen, and Xia 2023; Mirza et al. 2022; Khurana et al. 2021). On initial observation, it could be inferred that mini-batches, with fewer samples, would naturally produce flawed target statistics. However, this viewpoint is reductive. As depicted in Figure 1 (b)(c), the inaccurate estimation and performance degradation is efficiently reduced by increasing the class diversity within mini-batches. Our investigation indicates that *the quality of target estimation is not only influenced by the sample count but, more fundamentally, by the diversity within those samples.* Distortions arising from this discrepancy, which is independent of domain traits, could further obscure the true domain shifts and compromise model performance.

To tackle the discrepancies between training and testing phases in class diversity, we introduce a straightforward remedy, termed Test-time Exponential Moving Average (TEMA). TEMA utilises past statistics from previous batches, using a weighted average to enrich the diversity of class information in the current batch's statistics. However, as the data scope of TEMA expands (anticipated increase in class diversity), the model starts to lose its precise grasp on the current batch's data, leading to unstable and disrupted normalization. This gives rise to a trade-off: while concrete local context may suppress information richness, a focus on the global aspect compromises information timeliness. Recognizing this dilemma, we devise a versatile strategy that dynamically tailors the momentum, a parameter regulating TEMA's scope, to strike a balance between information richness and timeliness. This design broadens the scope of standard TTA methods beyond the current batch. By properly incorporating a diverse set of class information, TEMA stands out as a powerful tool for accurate target estimation irrespective of batch size at test time.

Built on the improvements from TEMA, one may directly replace source statistics with the target, which would risk model instability due to ever-changing parameters in BN layers, especially for complex tasks (see Table 4). While source statistics may not be ideally suited for challenges in target domains, they are instrumental in maintaining the model's stability and robustness — attributes often lacking in target data. In response to this, we propose a novel layer-wise rectification strategy that compensates target statistics with the source. Specifically, we leverage the divergence between target and source distributions as a guiding metric to adjust their contributions to our final normalization statistics. A significant divergence mandates a heavier reliance on the source to stabilize model performance, whereas a smaller divergence calls for a greater emphasis on the target to pinpoint the domain shifts. Our method effectively balances model stability with the use of BN to accurately capture domain shifts, enhancing overall test-time performance. Notably, our strategy stands out from traditional methods as it eliminates the need for manual parameter tuning, making it far more practical in real-world scenarios.

The contributions are summarized as follows:

- We investigate the underlying cause of inaccurate statistics during test-time adaptation, pinpointing the issue to the limited diversity of classes within batches.
- We introduce the Test-time Exponential Moving Average with an adaptive momentum mechanism. This approach dynamically balances the diversity of class information while ensuring timely updates.
- We propose a novel layer-wise normalization rectification strategy, considering the distribution divergence, to promote overall test-time performance.
- Extensive experiments exhibit consistent improvement and demonstrate remarkable stability.

Related Work

Test-time Adaption

A common challenge deep neural networks face is a decrease in performance when training and testing data present divergent distributions. To address this issue, substantial efforts (You et al. 2019; Cao et al. 2019; Hoffman et al. 2018) have been directed towards bridging the performance gap. Recently, Test-time adaptation (TTA) has emerged as a solution to combat the distribution shift from source to target domains during testing (Wang et al. 2021; Sun et al. 2020; Iwasawa and Matsuo 2021; Liang, Hu, and Feng 2020). Predominantly, these methods fall into two main categories: Test-time Training (TTT) (Sun et al. 2020; Liu et al. 2021; Gandelsman et al. 2022) and Fully Test-Time Adaptation (FTTA) (Wang et al. 2021; Niu et al. 2023; Zhang, Levine, and Finn 2022). TTT involves updating model parameters during testing and necessitates a specific auxiliary task during training, while FTTA presents a more challenging and realistic task by requiring the model to adapt online to the test stream without any modifications during training. This approach calls for the model to execute swift, real-time adjustments to effectively interpret and respond to the incoming data stream. In this paper, we direct our focus towards FTTA over TTT, as we are motivated by its capacity to meet the practical needs of dynamic and continuously evolving data environments.

Batch Normalization in Test-time Adaptation

Batch normalization (BN) (Ioffe and Szegedy 2015) has been widely used in deep neural networks for stable training and fast convergence. Recent TTAs mainly center on exploring the connections between batch normalization statistics and domain characteristics. One approach attempts to optimize the model during testing using target batch statistics and a specific loss function (Wang et al. 2021; Niu et al. 2022, 2023). Yet, this method neglects the challenge of accurately estimating the target distribution as the test batch size decreases, resulting in model deterioration under such circumstances. Another stream considers training-free methods by calibrating the normalization statistics. Since Nado et al. (2020) suggested prediction-time BN, which purely uses target statistics for standardization, Schneider et al. (2020) and

You, Li, and Zhao (2021) proposed to modify BN statistics by mixing the source and target via a predefined hyperparameter to mitigate the intermediate covariate shift. The latter two realized the drawbacks of target statistics and tried to cover up the inaccuracies with the source statistics, while leaving the persistent problem unresolved.

It is noted that recent work, TTN (Lim et al. 2023), bears some resemblance to ours, as both aim to address this inaccuracy issue. However, our approaches differ significantly in the solutions they employ. TTN augments source data to simulate test-time domain shifts and requires post-training to set a fixed balance between source and target statistics in final normalization. This strategy, while effective under certain conditions, demands extra training and lacks adaptability to varying target batches. In contrast, our method directly captures target statistics reflecting the intrinsic domain shifts, eliminating extra training. During inference, we adaptively balance between source and target weights for each incoming batch based on inter-domain divergence. This design enhances practicality in complex and diverse environments.

Test-time Exponential Moving Average

In previous studies, the Exponential Moving Average (EMA) mechanism was solely utilized during training to record more generalized source statistics with global information in BN layers. Recently, Zhao, Chen, and Xia (2023) and Yuan, Xie, and Li (2023) introduced EMA during the test phase to stabilize target normalization statistics, which are particularly prone to inaccuracies under mini-batch conditions. This process was coined as TEMA. Despite their improvements, the authors neither analyzed the root cause of the inaccurate target statistics nor clarified why TEMA was effective at mitigating this issue. In stark contrast, our in-depth analysis reveals the essence of this problem, identifying that the substantially reduced class diversity in target batches (compared to source batches) is the primary contributor to these inaccuracies. Motivated by this insight, we not only explain why TEMA can improve target estimation, but also refine it with an adaptive momentum mechanism to properly incorporate a diverse set of class information.

Preliminary

Problem Setting

In full test-time adaptation (TTA), we work with a parameterized model, $q_\theta(y|x)$, originally trained on a labeled source dataset $\mathcal{D}_s = \{(x, y) \sim p_s(x, y)\}$, where $x \in \mathcal{X}$ is the input and y is the ground truth label from the source class set $\mathcal{Y}$. We aim to improve the performance of this existing model during inference time for a continually changing target domain $\mathcal{D}_t = \{(x, y) \sim p_t(x, y)\}$ in an online fashion, without access to any source data and target labels. Note that for source and target distribution, $p_s(x, y) = p_s(x)p_s(y|x)$ and $p_t(x, y) = p_t(x)p_t(y|x)$, while covariate shift exists as $p_s(y|x) = p_t(y|x)$ and $p_s(x) \neq p_t(x)$. In this situation, the pretrained model $q_\theta(y|x)$ cannot closely represent the true, domain-invariant distribution $p(y|x)$. Therefore, TTA methods concentrate on adjusting $q_\theta(y|x)$ to maximize its predictive performance on the target distribution.

Batch Normalization

Batch normalization (BN) has been widely used in contemporary DNNs. During training, given a mini-batch $\mathcal{B} = \{x_n\}_{n=1}^N$ where $x_n \in \mathbb{R}^F$ is a feature vector (with F denoting the number of feature channels and N the batch size), BN normalizes each feature dimension f as follows:

$$\hat{x}_{n,f} = \frac{x_{n,f} - \mu_{\mathcal{B},f}}{\sqrt{\sigma_{\mathcal{B},f} + \epsilon}} \cdot \gamma_f + \beta_f, \tag{1}$$

where $\mu_{\mathcal{B},f}$ and $\sigma_{\mathcal{B},f}$ are the running mean and variance for the f-th feature of mini-batch $\mathcal{B}$, respectively. The parameters γ_f and β_f are the learned scale and shift factors for affine transformation, with ϵ being a small-offset to avoid division by zero. Meanwhile, the running mean $\mu_\mathcal{B} \in \mathbb{R}^F$ and covariance $\sigma_\mathcal{B} \in \mathbb{R}^F$ are accumulated to estimate the overall mean $\mathrm{E}[X]$ and covariance $\mathrm{Var}[X]$ of the training data:

$$\mathrm{E}[X] \Leftarrow m \cdot \mu_\mathcal{B} + (1 - m) \cdot \mathrm{E}[X] \,,$$
$$\mathrm{Var}[X] \Leftarrow m \cdot \sigma_\mathcal{B} + (1 - m) \cdot \mathrm{Var}[X] \,.$$

During inference, the conventional method computes BN with the estimated mean $\mathrm{E}[X]$ and covariance $\mathrm{Var}[X]$ from source, while advanced TTA methods that focusing on adapting BN layers adopt a different approach. They utilize statistics computed directly from each test batch to mitigate potential distributional shifts at test-time.

Method

Motivations

To motivate our approach, we first take a closer look at the target batch statistics during test time, as illustrated in Figure 1. Our observations are as follows: **1)** For sufficiently large target batches (e.g., 200), the statistics stabilize and offer an accurate description of target features, thereby aiding in test-time training. **2)** As the batch size diminishes, the statistics derived from the test batch become highly volatile and often inaccurate. Many existing techniques falter under these conditions because predictions rely on real-time statistics. **3)** Enhancing class diversity enables our approach to notably boost model performance, even when working with mini-batches. Driven by our findings, we challenge the prevailing notion in TTA research that the accuracy of target estimation primarily depends on batch sample size. Instead, we explore the intricate relationship between normalization statistics and class diversity with the following proposition.

Proposition 1. *Given an infinite sample space where each sample is independently and identically distributed (i.i.d) with an equal probability of selection for each category. Let M denote the number of distinct categories contained within a given batch, and K be the category number in total. For a batch of size N, the expected number of unique categories (also referred to as category diversity) is given by:*

$$E(M|N) = \sum_{k=1}^{K} \left[k \cdot \frac{\mathbf{C}_{N-1}^{k-1} \mathbf{C}_K^k}{\mathbf{C}_{N+K-1}^{K-1}} \right], \tag{2}$$

where $\mathbf{C}$ denotes the combination symbol in Combinatorics.

Proposition 1 quantifies the relationship between batch size and class diversity. Taking CIFAR-10 dataset as an example with a training batch size $N_s = 128$ (Hendrycks et al. 2020), we observe $E(M_s|N_s) \approx 9.34$. In case of the test batch size $N_t = 200$ — the default evaluation setting in previous research, $E(M_t|N_t) \approx 9.57$, which is nearly equal to $E(M_s|N_s)$. Such large target batch are advantageous, as their class diversity closely mirrors that of source batches, thereby yielding accurate target statistics. In real-world scenarios, the test-time batch size can often be much smaller, as in $N_t = 2$, a substantial discrepancy arises: $E(M_t|N_t) \approx 1.82$ significantly less than $E(M_s|N_s)$. This discrepancy is not merely a numerical one – it reflects a fundamental divergence in data distribution that is independent of domain characteristics. This divergence further obscures the inherent domain shifts and results in model degradation.

Accurate Target Estimation with TEMA

Enhanced Class Diversity We aim to bridge the gap between training and testing in terms of class diversity with a simple yet effective tool, namely, Test-time Exponential Moving Averages (TEMA). Specifically, TEMA operates by gradually absorbing statistics from previous batches into current estimations using the following formulas:

$$\mu_i^{\text{ema}} = m \cdot \mu_i^{\text{batch}} + (1-m) \cdot \mu_{i-1}^{\text{ema}}, \tag{3}$$

$$\sigma_i^{\text{ema}} = m \cdot \sigma_i^{\text{batch}} + (1-m) \cdot \sigma_{i-1}^{\text{ema}}, \tag{4}$$

where i denotes batch index. $\{\mu_i^{\text{batch}}, \sigma_i^{\text{batch}}\}$ represents the statistics of current batch, and $\{\mu_i^{\text{ema}}, \sigma_i^{\text{ema}}\}$ is the updated parameters for test-time normalization. Here, m refers to the "momentum", a crucial parameter that controls the amount of information retained from previous mini-batches with values ranging from 0 to 1. Larger m emphasizes the current batch's statistics, while a lower value prioritizes the accumulated historical statistics.

At its core, TEMA is designed to extend the effective sample pool for the current estimation. This is achieved through a weighted average that integrates information across multiple batches, as specified in the following proposition.

Proposition 2. *Given the iterative rules of TEMA defined in Eq. 3 and 4, it yeilds the i-th term as a cumulative sum of the past batch statistics weighted by $w_0 = (1-m)^i$ for initial batch and $w_t = (1-m)^{(i-t)}m$ for $t = 1, 2, ..., i$. Let ϵ be a threshold defining the effective sample batch, such that only batches with a relative weight $w_t/w_i > \epsilon$ are included. Then, the expanded sample scope for statistical estimation in TEMA can be formally expressed as $\hat{N}_t = \lfloor \log_{1-m} \epsilon \rfloor \cdot N_t$.*

Proof. We begin with the observation that, with momentum $m \leq 1$, the weight of each historical batch decays exponentially towards zero as the gap from the current batch i increases. To filter out negligible impact, we introduce a threshold ϵ, such that only batches with a relative weight $w_t/w_i > \epsilon$ are deemed effective. Given the continual influx of batches, the criterion simplifies to $(1-m)^{(i-t)} > \epsilon$, leading to $t > i - \log_{1-m} \epsilon$. This condition delineates the impactful batches for TEMA. As $\log_{1-m} \epsilon$ is not always an integer, we apply the floor function $\lfloor \cdot \rfloor$ to identify the nearest integer not exceeding this value. Therefore, the effective batches can be counted by $\lfloor \log_{1-m} \epsilon \rfloor$, further establishing an expanded sample scope of $\hat{N}_t = \lfloor \log_{1-m} \epsilon \rfloor \cdot N_t$. □

This enlarged sample pool allows TEMA to seamlessly incorporate a diverse set of class information. Importantly, this is not a heuristic but a deliberate design choice, inspired by our quantitative analysis that reveals the pivotal relationship between batch size and class diversity (see Proposition 1).

Adaptive Momentum While expanding TEMA's coverage indeed enhances class diversity, it meanwhile risks losing focus on the current batch's data, which could lead to unstable and disrupted normalization. This situation creates a trade-off. Prioritizing the local context of the current batch allows the model to capture fine-grained details, but may suppress information richness. Conversely, emphasizing a broader, global context enriches class diversity at the cost of information timeliness. Recognizing this dilemma, a pioneering contribution of our work is to introduce a versatile strategy that dynamically tailors the momentum in TEMA to navigate this trade-off.

Our primary goal is to identify the optimal momentum that balances two competing objectives: 1) aligning the enhanced class diversity during testing with that during training, and 2) keeping the enlarged sample pool as small as possible. We formalize this balance in the following optimization problem with a trade-off parameter $\lambda > 0$:

$$\arg\min_m \ |\frac{E(M_s|N_s)}{E(M_t|\hat{N}_t)} - 1| + \lambda \cdot \frac{\hat{N}_t}{N_s}, \tag{5}$$

where $\hat{N}_t = \lfloor \log_{1-m} \epsilon \rfloor \cdot N_t$ denotes the enlarged sample pool size (as defined in Proposition 2), and we respectively fix ϵ and λ at 0.1 and 0.01 for the sake of simplicity. To be specific, the first objective term quantifies the discrepancy between the class diversities during training and testing, while the second term severs a regularization that penalizes our enlarged sample pool size. Given that momentum is a standard hyper-parameter in deep learning, typically selected from the set $\{1, 0.1, 0.01, 0.001\}$, it would be inappropriate in our model to excessively optimize this parameter. We thus adopt a pragmatic approach: employing a grid search over this predefined set to seek the momentum value that minimizes our objective function.

Layer-wise Rectification Strategy

With TEMA's improvement on target estimation, one may be tempted to completely transition from source to target statistics in batch normalization (BN) layers. However, this may be counterproductive due to the inherent trade-off between target and source information. Target statistics are pivotal for proper normalization and domain shift alleviation, but they can also render the model unstable due to non-stationary parameters. On the other hand, source statistics, while not ideally suited for target domains, play a crucial role in preserving the model's stability and robustness.

To navigate this complex landscape, we propose a novel layer-wise rectification strategy that harmonizes the target statistics with the source. Central to this approach is the use of inter-domain divergence in each BN layer as a metric to

balance their contributions to the final normalization statistics. Our strategy is designed to be adaptive: it increases reliance on the source when the disparity between source and target distributions is significant, thereby promoting model stability. Otherwise, when the domain differences are more nuanced, our approach prioritizes the target statistics, ensuring precise adaptation to subtle domain shifts.

The overall pipeline of the proposed strategy is detailed in Algorithm 1, where we perform test-time batch normalization using combined statistics according to:

$$\mu^{(l)} = \alpha^{(l)} \cdot \mu_s^{(l)} + (1 - \alpha^{(l)}) \cdot \mu_t^{(l)} \ , \tag{6}$$

$$(\sigma^{(l)})^2 = \alpha^{(l)} \cdot (\sigma_s^{(l)})^2 + (1 - \alpha^{(l)}) \cdot (\sigma_t^{(l)})^2 + \alpha^{(l)} \cdot (1 - \alpha^{(l)})(\mu_s^{(l)} - \mu_t^{(l)})^2 \ . \tag{7}$$

In this context, $\alpha^{(l)}$ serves as a trade-off coefficient and is tied with inter-domain distribution divergence. $\{\mu_s^{(l)}, \sigma_s^{(l)}\}$ denote the source statistics for l-th BN layer, $\{\mu_t^{(l)}, \sigma_t^{(l)}\}$ refers to the target statistics estimated by TEMA.

One should note that in previous research, α is a manually tuned hyper-parameter for different tasks and is typically fixed for different incoming samples. While our strategy stands out from traditional ones by eliminating hyper-parameter tuning, enhancing the applicability of our approach in diverse real-world scenarios.

Algorithm 1: Layer-wise Rectification Strategy

Require: Test step $T := 0$; test stream sample D_{test}; Source pretrained model $q_\theta(y|x)$ with source statistics $\{\mu_s^{(l)}, \sigma_s^{(l)}\}$ for each l-th BN layer; Global prior $\mathcal{A}_T^{'} = [\alpha_T^{(1),\text{ema}}, \alpha_T^{(2),\text{ema}}, ..., \alpha_T^{(L),\text{ema}}]$ initialized with $\alpha_0^{(l),\text{ema}} = 0$, L denotes the BN layer number.

while the test batch arrives **do**
  // Stage 1: Model the source and target distribution and calculate divergence.
  for each BN layer l **do**
    Calculate the batch statistics $\mu_t^{(l)}, \sigma_t^{(l)} \in \mathbb{R}^F$.
    Combine the normalization statistics with $\alpha^{(l)} = \mathcal{A}_T^{'}[l]$ as Eq. (6) and Eq. (7).
    Model the distributions as MultiNormal Gaussians: $p_s^{(l)} = \mathcal{N}(\mu_s^{(l)}, \sigma_s^{(l)})$, $p_t^{(l)} = \mathcal{N}(\mu_t^{(l)}, \sigma_t^{(l)})$.
    Calculate the distribution divergence as: $D_{\text{KL}}^{(l)}(p_s^{(l)}, p_t^{(l)}) = 1/2 \cdot p_s^{(l)} \log\left(p_s^{(l)}/p_t^{(l)}\right) + 1/2 \cdot p_t^{(l)} \log\left(p_t^{(l)}/p_s^{(l)}\right)$.
  end for
  // Stage 2: Obtain the relative divergence $\mathcal{A}$.
  Normalization the divergence over L layers, Clip the value to [-1,1], and scale to [0,1]: $\mathcal{D} = [D_{\text{KL}}^{(1)}, D_{\text{KL}}^{(2)}, ..., D_{\text{KL}}^{(L)}]$. $\mathcal{A} = \gamma \cdot (\texttt{Clip}(\texttt{Norm}(\mathcal{D}) + 1)/2$. // γ is set as $1/2$ for emphasizing the target part.
  // Stage 3: Prediction.
  Combine the normalization statistics with $\alpha = \mathcal{A}$ and obtain final predictions.
  // Stage 4: Update the global prior $\mathcal{A}_{T+1}^{'}$.
  $\mathcal{A}_{T+1}^{'} = \tau \cdot \mathcal{A} + (1 - \tau) \cdot \mathcal{A}_T^{'}$ .// τ is set as 0.1.
  $T+ = 1$.
end while

Experiments and Analysis

Datasets and Model Architectures

We evaluate our approach on CIFAR-10-C, CIFAR-100-C, and ImageNet-C (Hendrycks and Dietterich 2018), which were initially designed to benchmark robustness of classification networks. All the corruption datasets are obtained by applying 15 kinds of corruption with 5 different degrees of severity on their clean test images of original datasets. CIFAR-10/CIFAR-100 originally has 10,000 test images, and ImageNet has 50,000 test images, which fall into 10/100/1000 categories, respectively. This results in a total of 150,000 test data for CIFAR-10-C/CIFAR-100-C, and 750,000 test data for ImageNet-C for each severity level.

Following the previous methods (Wang et al. 2021, 2022), we obtain the pretrained model from RobustBench benchmark (Croce et al. 2021), including the WildResNet-28 (Zagoruyko and Komodakis 2016) for CIFAR-10-C, ResNeXt-29 (Xie et al. 2017) for CIFAR-100-C (both pretrained by Hendrycks et al. (2020)), and ResNet-50 (He et al. 2016) for ImageNet-C (standard pretrained). All experiments are conducted on an RTX-3090 GPU.

Evaluation Settings

To show that our method performs robust on various test batch sizes, we conduct experiments with test batch sizes of 200, 64, 16, 4, 2, and 1 for CIFAR-10/100-C, and 64, 16, 4, 2, and 1 for ImageNet-C. Note that 200 for CIFAR-10/100-C and 64 for ImageNet-C are the widely-used evaluation batch size. Following Döbler, Marsden, and Yang (2023), we evaluate our methods in the following settings:

Continual: The model adapts to a sequence of test domains in an online manner without knowing when it changes.

Mixed domains: The model adapts to one long test sequence where consecutive test samples are likely to originate from different domains.

Gradual: The model adapts to a sequence of gradually increasing/decreasing domain shifts. This is formulated as: $\underbrace{...2}_{\text{type } t-1} \xrightarrow[\text{change}]{\text{type}} \underbrace{1 \to 2 \to 3 \to 4 \to 5 \to 4 \to 3 \to 2 \to 1}_{\text{corruption } t\text{, changing severity}} \xrightarrow[\text{change}]{\text{type}} \underbrace{1 \to 2...}_{\text{type } t+1}$

Compared Methods

We compare the following state-of-the-art training-free (TF) methods. TBN (Nado et al. 2020) re-estimates the batch normalization statistics from the test data. α-BN (You, Li, and Zhao 2021) combines the source and the test batch statistics with a pre-defined hyperparameter. AdaptiveBN (Schneider et al. 2020) proposes to reduce the covariate shift with the interpretation for normalization statistics of a N pseudo sample size for samples from the training set. Different from them, LAME (Boudiaf et al. 2022) shifted the focus from the model's parameters to the output probabilities via laplacian adjusted maximum-likelihood estimation.

Moreover, we also include recent training-required (TR)

Continual		CIFAR-10-C							CIFAR-100-C						
		200	64	16	4	2	1	Avg.	200	64	16	4	2	1	Avg.
	Source	43.50	43.50	43.50	43.50	43.50	43.50	43.50	46.45	46.45	46.45	46.45	46.45	46.45	46.45
TR	TENT	19.55	26.32	74.07	85.07	88.69	90.00	63.95	61.12	86.49	96.07	98.40	98.79	99.00	89.98
TR	CoTTA	16.24	17.65	34.36	78.88	87.79	90.00	54.15	32.68	34.30	47.52	92.62	97.84	98.96	67.32
TR	SAR	20.40	20.74	22.89	31.35	40.32	89.83	37.59	31.90	35.89	54.84	66.08	73.24	98.91	60.14
TR	AdaCont.	18.50	17.41	19.69	35.01	63.81	31.55	31.00	33.61	35.40	55.04	89.45	96.16	62.67	62.06
TR	ETA	17.64	20.01	30.60	56.78	83.24	89.83	49.68	32.31	35.17	44.72	88.22	98.96	98.91	66.38
TF	TBN	20.35	20.82	23.06	31.62	38.57	89.83	37.38	35.50	36.29	39.67	52.73	73.24	98.91	56.06
TF	α-BN	30.60	30.67	30.89	31.89	32.91	34.47	31.91	37.02	38.27	35.92	37.17	37.25	41.18	37.80
TF	AdaptBN	20.36	20.71	21.98	26.79	32.19	37.52	26.59	35.40	**35.78**	35.92	37.14	39.23	41.85	37.55
TF	LAME	64.52	57.66	47.70	44.38	43.75	90.00	58.00	98.49	73.83	47.59	46.64	46.50	99.00	68.68
	Ours	**20.20**	**20.57**	**20.74**	**21.45**	**20.91**	**21.05**	**20.82**	**34.63**	36.11	**35.31**	**36.02**	**36.32**	**39.30**	**36.28**

Table 1: Continual adaptation on corruption benchmark CIFAR-10-C/CIFAR-100-C. Error rate (↓) averaged over 15 corruptions with severity level 5 for each test batch size (200/64/16/4/2/1).

Continual		ImageNet-C					
		64	16	4	2	1	Avg.
	Source	82.00	82.00	82.00	82.00	82.00	82.00
TR	TENT	62.60	91.99	99.74	99.85	99.90	90.82
TR	CoTTA	63.10	84.35	99.79	99.88	99.79	89.38
TR	SAR	68.04	62.62	79.19	92.71	99.19	80.35
TR	AdaCont.	66.83	92.02	98.40	99.66	99.88	91.36
TR	ETA	58.68	69.65	98.81	93.14	99.19	83.89
TF	TBN	68.60	70.79	83.10	92.74	99.22	82.89
TF	α-BN	**63.38**	**63.75**	64.98	66.61	68.90	**65.53**
TF	AdaptBN	64.13	65.41	65.63	66.72	**68.12**	66.00
TF	LAME	93.50	73.59	73.22	73.18	99.90	82.68
	Ours	64.15	64.90	**64.87**	**66.01**	68.79	65.74

Table 2: Continual adaptation on ImageNet-C.

methods for reference. TENT (Wang et al. 2021) focuses on optimizing batch normalization by minimizing the entropy of the model's predictions. The study of CoTTA (Wang et al. 2022) delves into long-term test-time adaptation in environments that continually change. Methods like ETA (Niu et al. 2022) and SAR (Niu et al. 2023) are geared towards excluding samples that are deemed unreliable and redundant from the optimization process. AdaContrast (Chen et al. 2022) leverages contrastive learning to enhance feature learning, incorporating a mechanism for refining pseudo-labels.

Results

Table 1,2 show the error rates on three corruption benchmark datasets under a continual evaluation setting. In Table 3, we report the error rates on CIFAR-10-C under mixed-domain adaptation and gradual changing shifts, respectively.

Our method consistently outperforms other approaches. As presented in Table 1,2, our method shows a robust and superior performance on all corruption benchmark datasets. The results highlight the consistency of our method in delivering the lowest error rates. Notably, on the CIFAR-10-C dataset, our approach achieved an average error rate of 20.82% across all batch sizes. This is remarkably lower compared to the next-best performing method, AdaptBN, with an average error rate of 26.59%.

Our method maintains a high level of performance even with the reduction of batch size. The robustness of our method is further illustrated by the minimal increase in error rates as the batch size decreases. For example, on the CIFAR-10-C dataset, the error rate increases marginally from 20.20% with a batch size of 200 to 21.05% with a batch size of 1 (Table 1). While for TENT, CoTTA and other training-required methods, this degradation is significant due to the dependence on the inaccurate test batch statistics. Note that compared with α-BN and AdaptBN which leverage large source statistics (80% or more under mini-batches), their performance are limited with mini-batches though they do avoid degradation.

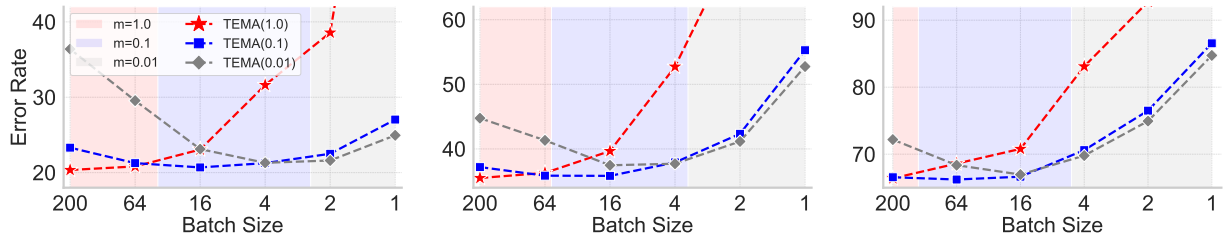
Our method demonstrates remarkable stability across different corruption levels and various practical scenarios. The results from Table 3 underscore our method's robustness in diverse (mixed domain) and changing (gradual) environments. Our method still stands out with its strong performance, especially under small batch sizes – conditions that can challenge many methods. While other methods tend to display considerable fluctuations in error rates across different corruption levels (especially for the optimization-based methods), our approach manages to maintain relatively steady performance. The solid empirical results validate the proposed approach's robustness and adaptability, highlighting its potential value for real-world deployment where these types of changes are commonplace.

Momentum Analysis for TEMA

We conduct experiments using benchmark datasets to evaluate the proposed adaptive momentum strategy. The process is visualized in Figure 2. Note that the intersections of the lines do not imply the two methods exhibit identical performance under this configuration. Instead, they represent trends for different methods, serving to illustrate our estimations correspond to the actual performance observed under the tested batch size. The results not only showcase the congruence between our estimated values and empirical perfor-

	CIFAR-10C	Mixed Domain							Gradual						
		200	64	16	4	2	1	Avg.	200	64	16	4	2	1	Avg.
	Source	43.50	43.50	43.50	43.50	43.50	43.50	43.50	24.66	24.66	24.66	24.66	24.66	24.66	24.66
TR	TENT	39.75	57.14	80.30	88.39	89.26	90.00	74.14	26.32	56.19	81.79	90.84	87.96	90.00	72.18
TR	CoTTA	32.37	31.22	51.30	85.04	86.97	89.85	62.79	11.16	16.83	59.38	86.96	89.68	90.00	59.00
TR	SAR	33.74	33.97	35.38	41.13	48.86	89.83	47.15	13.97	14.43	16.51	25.28	32.47	89.75	32.07
TR	AdaCont.	26.12	23.85	23.58	35.57	62.50	46.01	36.27	12.33	13.66	19.93	33.13	54.17	22.33	25.93
TR	ETA	27.92	34.81	55.07	69.62	85.43	89.83	60.45	16.63	21.82	57.51	77.46	84.59	89.75	57.96
TF	TBN	**33.77**	**34.05**	35.51	40.72	45.05	89.83	46.49	13.64	14.10	16.21	25.10	31.66	89.75	31.74
TF	α-BN	39.85	39.85	39.59	38.80	37.46	**34.47**	38.34	18.05	18.12	18.25	18.87	19.49	20.13	18.82
TF	AdaptBN	33.86	34.35	35.69	36.96	37.10	37.52	35.91	13.63	13.96	14.86	17.20	19.23	21.57	16.74
TF	LAME	75.09	52.30	44.87	43.74	43.55	90.00	58.26	34.22	31.21	26.69	25.13	24.84	90.00	38.68
	Ours	34.18	34.35	**34.48**	**34.79**	**35.59**	37.33	**35.12**	**13.50**	**13.83**	**14.03**	**14.38**	**13.71**	**13.99**	**13.91**

Table 3: Mixed Domain/ Gradual adaptation on corruption benchmark CIFAR-10-C. For gradual setting, error rate (↓) averaged over 15 corruptions and severity level 1-5.



(a) CIFAR-10-C (b) CIFAR-100-C (c) IMAGENET-C

Figure 2: Momentum analysis for TEMA on three benchmarks under continual setting with different test batch size. Red, blue and grey regions represent the calculated part where momentum should be set to $m = 1.0, 0.1, 0.01$ according to Eq. (5). Lines plot the experimental performance of TEMA(m=1.0)/TBN, TEMA(m=0.1), and TEMA(m=0.01).

	Method	Test batch size						
		200	64	16	4	2	1	Avg.
1	Baseline	43.50	43.50	43.50	43.50	38.57	43.50	43.50
1	TBN	20.35	20.82	23.06	31.62	38.57	89.83	37.38
1	TEMA	20.35	20.82	**20.69**	**21.26**	21.61	24.95	21.61
1	Ours	**20.20**	**20.57**	20.74	21.45	**20.91**	**21.05**	**20.82**
2	Baseline	46.45	46.45	46.45	46.45	46.45	46.45	46.45
2	TBN	35.50	36.29	39.67	52.73	73.24	98.91	56.06
2	TEMA	35.50	**35.86**	35.83	37.71	41.16	52.75	39.80
2	Ours	**34.63**	36.11	**35.31**	**36.02**	**36.32**	**39.30**	**36.28**
3	Baseline	82.00	82.00	82.00	82.00	82.00	82.00	82.00
3	TBN	66.40	68.60	70.79	83.10	92.74	99.22	82.89
3	TEMA	66.40	66.22	66.64	69.79	74.94	84.74	72.47
3	Ours	**64.05**	**64.15**	**64.90**	**64.87**	**66.01**	**68.79**	**65.46**

Table 4: Ablation study on three corruption benchmarks (1 → CIFAR-10-C, 2 → CIFAR-100-C, 3 → ImageNet-C).

mance but also highlight our achievement in effectively balancing the class diversity and timeliness of target statistics.

Ablation Study

We conduct an ablation study on the importance of the proposed method under the continual setting. The results are shown in Table 4. The Baseline model displays consistent error rates across various test batch sizes. However, the introduction of target normalization statistics (TBN/TEMA) to the model significantly reduced the error rates across all test batch sizes. While with TEMA, an optimal selection of the momentum for target statistics, this component yields robust improvements over all batch sizes, reaching the lowest error rate. Finally, the addition of Layer-wise Rectification Strategy (Ours) leads to the best performance across all batch sizes, showcasing the benefit of a layer-wise normalization rectification strategy for better generalization ability.

Conclusions and Limitations

This paper examines test-time degradation by unraveling batch normalization, identifying the reduced class diversity in batches as the key issue. To mitigate this problem and promote test-time performance, we 1) introduce TEMA with adaptive data scope, designed to bridge the class diversity gap between training and testing, and 2) propose a novel layer-wise rectification strategy, calibrated using inter-domain divergence to harmonize source and target statistics. Experiments in diverse real-world scenarios demonstrate the superiority of our method in comparison with the state-of-the-arts. While our proposed strategy shows effectiveness in most scenarios, it presumes samples to be i.i.d, which may not apply to all circumstances. Future work could involve extending the exploration to non-i.i.d settings.

Appendix

Proof of Proposition 1

Proof. We begin with the premise of an i.i.d. sample space with K equally probable categories. For any given $k < K$, the combinations $\mathbf{C}_K^k$ represent the ways to select k categories from K. After choosing k classes, the arrangements where each category gets at least one sample in a batch of N samples are counted by $\mathbf{C}_{N-1}^{k-1}$, and the total number of ways to assign N samples into $1, 2, \ldots, K$ classes is $\mathbf{C}_{N+K-1}^{K-1}$. As such, the probability of having exactly k unique categories in a batch is the ratio of these combinatory figures. Finally, we can derive the expected category diversity by summing all unique categories k times their respective probabilities as $E(M|N) = \sum_{k=1}^{K} \left[k \times \frac{\mathbf{C}_{N-1}^{k-1} \mathbf{C}_K^k}{\mathbf{C}_{N+K-1}^{K-1}} \right]$. □

Flowchart of Layer-wise Rectification Strategy

We present a chart in Figure 3 to visually illustrate the procedure of Algorithm 1.

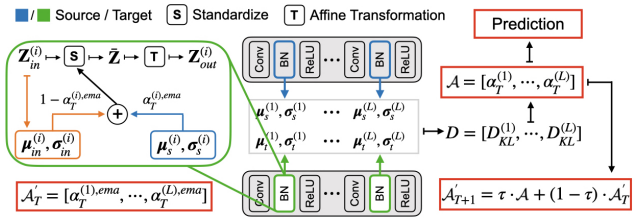
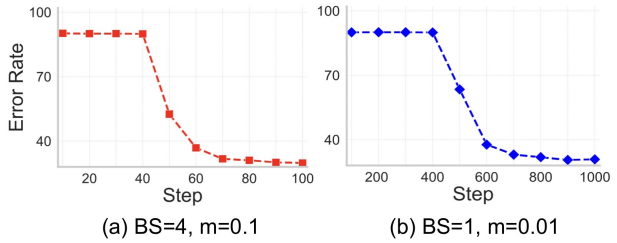


Figure 3: Flowchart of Layer-wise Rectification Strategy.

Model progression with data expansion

We also examine the relationship between real-time model performance and the volume of incoming data in Figure 4 and observe that the adaptation outcome may not be satisfactory during the initial phase. This likely stems from TEMA's early struggles with target distribution estimation, thereby undermining the calibration parameter α and ultimate model performance. This observation aligns with our stance in the main text that the limited class diversity (in initial TEMA phase) would compromise model performance. Given our primary focus on online Test-Time Adaptation, the issue of total data volume did not initially receive significant attention. We will delve deeper into this problem in future work.



(a) BS=4, m=0.1 (b) BS=1, m=0.01

Figure 4: Real-time performance on CIFAR-10-C (Gaussian noise). Error rate remains stable for steps after.

Acknowledgements

This work was supported by: National Natural Science Foundation of China under No. 92370119, No. 62376113,No. 62276258, and No. 62206225; Jiangsu Science and Technology Program (Natural Science Foundation of Jiangsu Province) under No. BE2020006-4; Natural Science Foundation of the Jiangsu Higher Education Institutions of China under No. 22KJB520039.

References

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*, 8344–8353.

Cao, Z.; You, K.; Long, M.; Wang, J.; and Yang, Q. 2019. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2985–2994.

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*, 295–305.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 834–848.

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*.

Döbler, M.; Marsden, R. A.; and Yang, B. 2023. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7704–7714.

Gandelsman, Y.; Sun, Y.; Chen, X.; and Efros, A. 2022. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35: 29374–29385.

Ganin, Y.; and Lempitsky, V. 2015. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 1180–1189. PMLR.

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. G. 2018. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. 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*, 1989–1998. Pmlr.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 448–456. PMLR.

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.

Khurana, A.; Paul, S.; Rai, P.; Biswas, S.; and Aggarwal, G. 2021. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Liang, J.; Hu, D.; and Feng, J. 2020. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, 6028–6039. PMLR.

Lim, H.; Kim, B.; Choo, J.; and Choi, S. 2023. TTN: A Domain-Shift Aware Batch Normalization in Test-Time Adaptation. In *International Conference on Learning Representations*.

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? *Advances in Neural Information Processing Systems*, 34: 21808–21820.

Long, M.; Wang, J.; Ding, G.; Sun, J.; and Yu, P. S. 2013. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2200–2207.

Mirza, M. J.; Micorek, J.; Possegger, H.; and Bischof, H. 2022. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14765–14775.

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. *arXiv preprint arXiv: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*, 16888–16905. PMLR.

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. In *International Conference on Learning Representations*.

Schneider, S.; Rusak, E.; Eck, L.; Bringmann, O.; Brendel, W.; and Bethge, M. 2020. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33: 11539–11551.

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*, 9229–9248. PMLR.

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

Wang, Q.; Fink, O.; Van Gool, L.; and Dai, D. 2022. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7201–7211.

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.

You, F.; Li, J.; and Zhao, Z. 2021. Test-time batch statistics calibration for covariate shift. *arXiv preprint arXiv:2110.04065*.

You, K.; Long, M.; Cao, Z.; Wang, J.; and Jordan, M. I. 2019. Universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2720–2729.

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*, 15922–15932.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, M.; Levine, S.; and Finn, C. 2022. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35: 38629–38642.

Zhao, B.; Chen, C.; and Xia, S.-T. 2023. DELTA: Degradation-free Fully Test-time Adaptation. In *International Conference on Learning Representations*.