# On-Device Realistic Test-Time Adaptation via Bias-Resistant Statistical Alignment

**Haojie Bai**[1,2], **Aiguo Chen**[1,2*], **Ruiting Dai**[1,2], **Yijia Rong**[1,2], **Zirui Wang**[1,2], **Jiaxin Liu**[1,2], **Kexin Li**[1,2], and **Schahram Dustdar**[3,4]
[1]University of Electronic Science and Technology of China
[2]Ubiquitous Intelligence and Trusted Services Key Laboratory of Sichuan Province
[3]TU Wien
[4]Catalan Institution for Research and Advanced Studies
{202411900413, 202421900206, 202522900217, 202411900410}@std.uestc.edu.cn,
{agchen, rtdai, likx}@uestc.edu.cn, dustdar@dsg.tuwien.ac.at

## Abstract

Test-Time Adaptation (TTA) aims to adapt pre-trained models to unseen test data, which is crucial for resource-constrained edge devices that must handle distribution shifts on the fly without human supervision. However, conventional TTA methods often fail in realistic scenarios characterized by continuous shifts and severe class imbalances while incurring prohibitive memory overheads. To enable continuous adaptation to test data on edge devices under a realistic TTA setting, we propose a **B**ias-Resistant **O**nline **S**tatistical **A**lignment (**BOSA**) method. BOSA facilitates unbiased adaptation via a discrepancy-aware statistical alignment mechanism integrated with a class-balanced memory bank. To ensure memory efficiency, we further design a saliency-guided activation sparsification and gradient reconstruction scheme, which drastically reduces memory overhead without sacrificing gradient integrity. Extensive evaluations demonstrate that BOSA achieves superior accuracy with a compact memory consumption compared to state-of-the-art methods under realistic TTA settings.

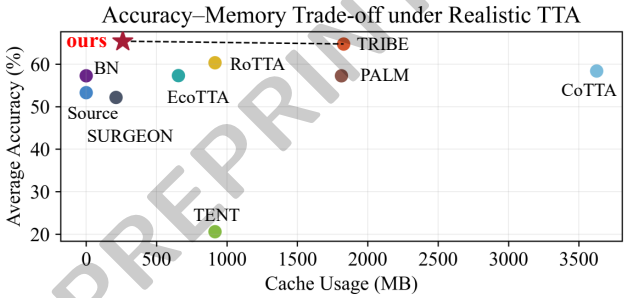



Figure 1: Accuracy vs. memory overhead trade-off under the realistic TTA setting. Our BOSA achieves the highest average accuracy while maintaining a substantially reduced cache footprint, outperforming existing state-of-the-art TTA methods in both performance and resource efficiency.

## 1 Introduction

Many real-world applications require deploying well-trained deep neural networks (DNNs) to edge environments. However, model performance often degrades substantially when the test data come from a distribution different from the training data, due to factors such as weather variations, diurnal cycles, and device movement [Wang *et al.*, 2022; Brahma and Rai, 2023; Yang *et al.*, 2024; Zhao *et al.*, 2025]. To address this critical problem, Test-Time Adaptation (TTA) has attracted increasing attention for adapting pre-trained models online to unlabeled test data streams with distribution shifts [Wang *et al.*, 2020; Su *et al.*, 2022; Yuan *et al.*, 2023; Su *et al.*, 2024a; Hu *et al.*, 2025b; Wang *et al.*, 2025].

Given that the fundamental goal of TTA is to enable unseen distribution adaptation in practical applications, an ideal TTA method is expected to operate effectively under typical testing environments. However, much of the prior work has been conducted under settings that substantially deviate from real-world constraints. Specifically, most existing TTA methods, which process the test data stream batch-by-batch, often impose the requirement that the data within each batch must be class-balanced [Wang *et al.*, 2020; Wang *et al.*, 2022; Song *et al.*, 2023; Hu *et al.*, 2025b; Maharana *et al.*, 2025]. This is an impractical consideration for realistic applications, as sufficiently class-balanced test data is unlikely to be available within short temporal horizons. Furthermore, most current TTA methods suffer from heavy storage overhead, as they overlook the reduction of activation for backpropagation, which serves as the primary memory bottleneck for on-device learning [Cai *et al.*, 2020; Hong *et al.*, 2023; Ma *et al.*, 2025].

A growing body of recent studies [Niu *et al.*, 2023; Yuan *et al.*, 2023; Hu *et al.*, 2025b] has begun to relax these class-balanced assumptions by considering a non-i.i.d. TTA setting, where test data exhibit local class imbalance and continual distribution shifts. Under such conditions, batch-dependent Batch Normalization (BN) statistics tend to drift toward majority classes, leading to degraded and unstable test-time adaptation. Beyond local imbalance, [Su *et al.*, 2024a; Jang and Chung, 2025] have investigated a realistic

*Contact Author

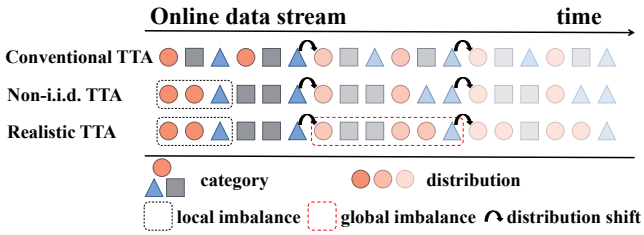



Figure 2: Conventional TTA assumes a continuously shifting test-time distribution over an online data stream. Building upon this, the non-i.i.d. setting further accounts for local class imbalance within individual batches. The realistic TTA setting considered here extends this formulation by additionally modeling global class imbalance over longer temporal horizons, which significantly exacerbates bias accumulation in normalization statistics.

TTA setting that additionally accounts for global class imbalance over longer temporal horizons. To highlight the differences between these TTA settings, we provide illustrative comparisons in Fig. 2. The realistic TTA setting poses substantially greater challenges, as persistent global class imbalance progressively amplifies the bias accumulation of BN statistics over time. Consequently, most existing TTA methods [Wang *et al.*, 2020; Wang *et al.*, 2022; Yuan *et al.*, 2023; Ma *et al.*, 2025; Maharana *et al.*, 2025] that depend on these statistics suffer from severe performance degradation, as illustrated in Fig. 1. Furthermore, most TTA methods entail substantial storage overhead (Fig. 1), which limits their deployment on resource-constrained edge devices. While some memory-efficient TTA approaches [Song *et al.*, 2023; Ma *et al.*, 2025] aim to reduce backpropagation activations, they are not specifically tailored for realistic TTA settings.

To mitigate the inherent bias of test data while reducing memory overhead, we propose **BOSA**: a **B**ias-resistant **O**nline **S**tatistical **A**lignment framework for realistic on-device test-time adaptation, underpinned by a saliency-guided activation sparsification and gradient reconstruction mechanism. Specifically, we align running statistics by characterizing the distributional discrepancy between the input features and the maintained statistics, complemented by a class-balanced memory bank designed to bypass the imbalanced inputs. To minimize memory consumption during adaptation, we selectively sparsify activations based on channel saliency and reconstruct the gradients of these sparsified activations to preserve adaptation effectiveness. Finally, a self-training strategy is employed to bolster model robustness and performance.

Our contributions are as follows.

- To the best of our knowledge, this is the first study to address Test-Time Adaptation (TTA) under the dual real-world constraints of imbalanced test data and memory efficiency. To this end, we propose BOSA, a Bias-resistant Online Statistical Alignment framework specifically designed for realistic on-device TTA.
- The proposed framework innovatively combines a discrepancy-aware alignment mechanism with a class-balanced memory bank to rectify running statistics, which ensures stable and unbiased adaptation even under severe class imbalances.
- It further incorporates a saliency-guided activation sparsification and gradient reconstruction mechanism during optimization, enabling memory-efficient backpropagation without underfitting model parameters.
- Extensive experiments on actual edge devices demonstrate that, compared to existing methods under realistic constraints, our approach achieves state-of-the-art accuracy with substantially reduced memory requirements, highlighting its suitability for real-world deployment.

## 2 Related Work

This section reviews the existing Test-Time Adaptation research and briefly introduces emerging directions, including realistic TTA and memory-efficient TTA.

### 2.1 Test-time Adaptation

Test-Time Adaptation aims to improve model performance on distribution-shifted test data. Existing methods can be broadly categorized into two types: 1) Test-Time Training (TTT): introduces auxiliary self-supervised objectives during training, enabling adaptation at test time by continuing to optimize these objectives [Hakim *et al.*, 2023; Deng *et al.*, 2025]. 2) Fully Test-Time Adaptation (FTTA): adapt models exclusively during inference without modifying the training pipeline [Wang *et al.*, 2020; Boudiaf *et al.*, 2022; Niu *et al.*, 2023; Tsai *et al.*, 2024; Liu *et al.*, 2025]. Our work focuses on the practical and widely studied FTTA paradigm.

### 2.2 Realistic TTA

State-of-the-art FTTA methods mainly rely on self-training [Wang *et al.*, 2022; Yuan *et al.*, 2023; Ma, 2024; Han *et al.*, 2025], distribution alignment [Wang *et al.*, 2023; Wang *et al.*, 2024; Adachi *et al.*, 2025], or entropy minimization [Wang *et al.*, 2020; Yuan *et al.*, 2023; Lee *et al.*, 2024; Tan *et al.*, 2025]. While effective under conventional TTA settings, most of these methods are developed under idealized assumptions and struggle in realistic deployment scenarios. In practice, non-i.i.d. test data streams with local class imbalance [Gong *et al.*, 2022; Yuan *et al.*, 2023; Niu *et al.*, 2023; Wang *et al.*, 2024; Niu *et al.*, 2025] pose substantial challenges to existing TTA methods. More recent work, such as TRIBE [Su *et al.*, 2024a], further considers persistent global class imbalance and achieves strong performance through tri-network self-training and balanced normalization. Despite their effectiveness, current realistic TTA methods largely overlook memory efficiency, which limits their applicability on resource-constrained edge devices.

### 2.3 Memory-efficient TTA

Orthogonal to the aforementioned research, memory-efficient TTA methods aim to reduce the prohibitive memory overhead during adaptation, enabling deployment on resource-constrained devices. These methods are motivated by the observation that memory consumption is dominated by storing intermediate activations for backpropagation [Cai *et al.*,

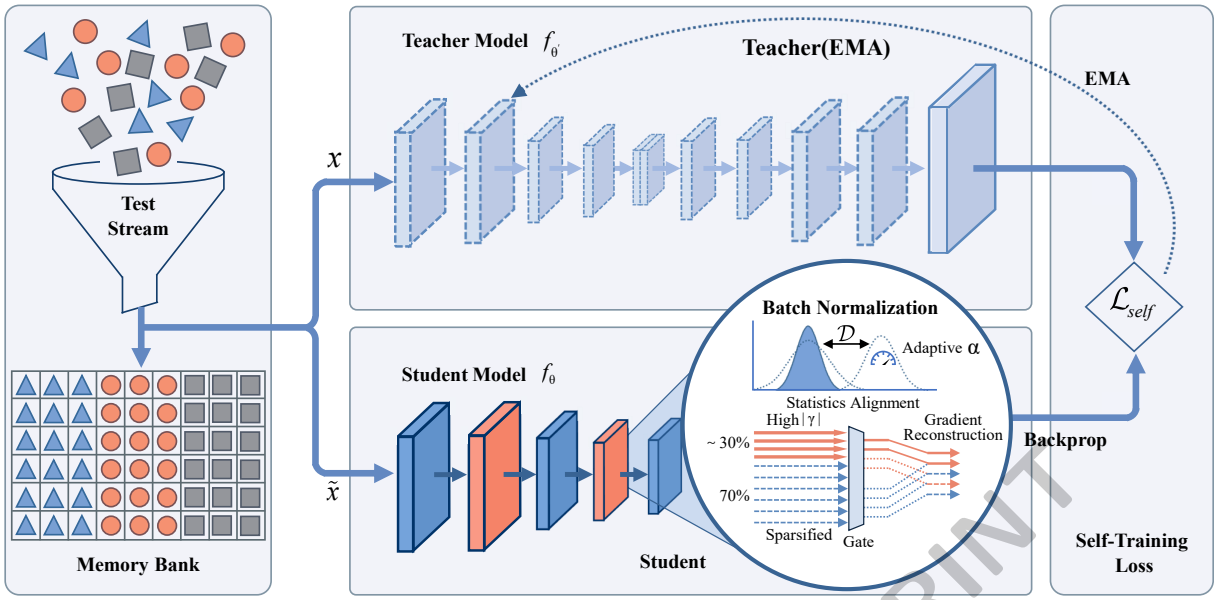



Figure 3: The framework of BOSA. Our method facilitates online adaptation via a teacher-student paradigm. Key innovations include discrepancy-aware statistical alignment and saliency-guided activation sparsification with gradient reconstruction, both of which refine the Batch Normalization process for bias-resistant statistical alignment and minimal memory consumption. The model maintains long-term robustness through self-training loss and EMA-based parameter updates.

2020]. Accordingly, existing approaches primarily aim to reduce the number of activations that need to be stored. Representative strategies include activation sparsification [Hong *et al.*, 2023; Niu *et al.*, 2024; Ma *et al.*, 2025; Hu *et al.*, 2025a], early-exit mechanisms [Jia *et al.*, 2024; Sahoo *et al.*, 2024; Li *et al.*, 2025], or lightweight branch updating that avoids adapting the full backbone [Song *et al.*, 2023; Shin and Kim, 2024; Chi *et al.*, 2025]. Despite their memory efficiency, these methods are primarily evaluated under conventional TTA settings and have not been systematically studied in realistic TTA scenarios.

## 3 Preliminaries for Realistic Test-Time Adaptation

Given a model $f_{\Theta_0}$ pretrained on a source domain $\mathcal{D}_S = (\mathcal{X}_s, \mathcal{Y}_s)$, the realistic test-time adaptation setting aims to adapt the model to a continuously arriving unlabeled data stream $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, ..., \mathcal{X}_T$, where at each time step $t$, a mini-batch $\mathcal{X}_t$ is sampled from the test distribution $\mathcal{P}^t_{test}$. Notably, $\mathcal{P}^t_{test}$ evolves, being jointly governed by both the underlying test domain and the class label distribution. In this setting, both the test domain and the class distribution vary continuously over time: the incoming test data exhibit local class imbalance over short temporal horizons and persistent global class imbalance over longer temporal horizons, across multiple gradual and smooth domain shifts. Following [Su *et al.*, 2024a], we construct the continuously arriving unlabeled test data stream $\{\mathcal{X}_t\}_{t=1}^T$ using a hierarchical sampling process. First, we simulate persistent global class imbalance within each test domain by enforcing a long-tailed class prior controlled by an imbalance factor $\rho$, where class frequencies decay exponentially. Next, we induce local class imbalance by partitioning the globally imbalanced samples into multiple temporal segments, with class proportions in each segment governed by a Dirichlet distribution parameterized by $\delta$.

## 4 Method

### 4.1 Overview

As illustrated in Figure 3, the proposed BOSA framework primarily comprises three collaborative modules: 1) a student-teacher self-training strategy tailored to adapt to realistic test streams; 2) a bias-resistant online statistical alignment strategy that aligns running statistics by characterizing distributional discrepancies; 3) a saliency-guided activation sparsification and gradient reconstruction mechanism designed to selectively reduce channel-wise activations and reconstruct missing gradients.

### 4.2 Self-Training Strategy

To enhance robustness on unlabeled test samples, we employ a self-training loss that enforces predictive consistency between the teacher $f_{\Theta'_0}$ and student $f_{\Theta_0}$ models. At each time step $t$, the incoming imbalanced mini-batch is evaluated by the teacher model and selectively incorporated into a class-balanced memory bank via low-entropy filtering. This memory bank serves to provide high-quality samples that approximate an i.i.d. distribution for the self-training objective. Specifically, these samples are fed into the teacher, while their augmented views are input to the student to update the student's affine BN parameters by minimizing the following self-training loss:

$$\mathcal{L} = \frac{1}{\mathcal{B}} \sum_{i=1}^{\mathcal{B}} \mathrm{CE}\Big( f_{\Theta'_t}(x_i),\ f_{\Theta_t}(\tilde{x}_i) \Big) \tag{1}$$

where $\mathrm{CE}(\cdot,\cdot)$ denotes the cross-entropy loss, $\mathcal{B}$ is the capacity of the memory bank, $x_i$ is the original sample retrieved from the memory bank, and $\tilde{x}_i$ is its corresponding augmented view. The teacher model is subsequently updated as an exponential moving average (EMA) of the student parameters: $\Theta'_{t+1} = (1-v_m)\,\Theta'_t + v_m\,\Theta_{t+1}$, where $v_m \in [0,1]$ is the momentum coefficient.

### 4.3 Bias-Resistant Online Statistics Alignment

The inherent class imbalance in realistic TTA streams often causes BN statistics $\{\mu, \sigma^2\}$ to drift toward majority classes. Consequently, we leverage the class-balanced memory bank introduced in Section 4.2 to decouple the statistics estimation from the biased input. Furthermore, we propose a discrepancy-aware statistical alignment mechanism to achieve unbiased statistics estimation for the student's BN layers under dynamic distributions $\mathcal{P}^t_{test}$, which is performed concurrently with the optimization process in Eq. 1.

Specifically, to align the running statistics of channel $c$ with the target distribution at iteration $t$, we adopt an Exponential Moving Average (EMA) strategy for online statistics estimation:

$$\mu_{t,c} = (1-\alpha)\mu_{t-1,c} + \alpha\hat{\mu}_{t,c}, \tag{2}$$

$$\sigma^2_{t,c} = (1-\alpha)\sigma^2_{t-1,c} + \alpha\hat{\sigma}^2_{t,c}, \tag{3}$$

where $\hat{\mu}_{t,c}$ and $\hat{\sigma}^2_{t,c}$ capture the statistics of the target distribution, derived from the intermediate features of samples retrieved from the memory bank, and $\alpha \in [0,1]$ controls the alignment rate. Intuitively, a larger $\alpha$ is essential to rapidly align the running statistics with the target distribution when significant shifts are detected. Conversely, $\alpha$ should be minimized to preserve acquired structural information. To formalize this intuition, we introduce a mechanism that adaptively calibrates $\alpha$ based on the quantified distribution discrepancies:

$$\alpha_t = 1 - e^{-\mathcal{D}\left(\mathcal{N}(\mu_{t-1},\sigma^2_{t-1}),\mathcal{N}(\hat{\mu}_t,\hat{\sigma}^2_t)\right)} \tag{4}$$

$\mathcal{N}(\hat{\mu}_t, \hat{\sigma}^2_t)$ and $\mathcal{N}(\mu_{t-1}, \sigma^2_{t-1})$ are two parameterized Gaussian distributions, representing the target distribution at iteration $t$ and the distribution characterized by the running statistics, respectively. $\mathcal{D}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2C}\sum_{i=1}^{C}\Big(\mathrm{KL}(\mathcal{N}_{1,i} \parallel \mathcal{N}_{2,i}) + \mathrm{KL}(\mathcal{N}_{2,i} \parallel \mathcal{N}_{1,i})\Big)$ denotes a discrepancies measure between the two distributions, where $C$ is the channel dimension. We adopt a symmetric KL divergence to avoid directional bias in discrepancy estimation. A larger distributional discrepancy $\mathcal{D}$ leads to a larger alignment coefficient $\alpha_t$.

While $\alpha_t$ ensures responsiveness to dynamic distributions, it may inadvertently lead the model to deviate excessively from the robust features learned in the source distribution. To mitigate the risk of overfitting to the target distribution, we further introduce an adaptive update constraint. Specially, we regularize $(\mu_t, \sigma^2_t)$ by minimizing the Wasserstein-2 distance between the $\mathcal{N}(\mu_t, \sigma^2_t)$ and the source distribution $\mathcal{N}(\mu_s, \sigma^2_s)$:

$$\begin{aligned}\mathcal{L}_{reg} &= \alpha_t \mathcal{W}_2^2\left(\mathcal{N}(\mu_t,\sigma^2_t), \mathcal{N}(\mu_s,\sigma^2_s)\right) \\ &= \alpha_t\left[(\mu_t-\mu_s)^2 + (\sigma_t-\sigma_s)^2\right]\end{aligned} \tag{5}$$

where $\alpha_t$ is the adaptive strength of the constraint derived from the distribution discrepancy in Eq. 4. Concretely, we perform a gradient-based refinement step following the EMA alignment (as described in Eqs. 2 and 3). By computing the partial derivatives of $\mathcal{L}_{reg}$ with respect to the running statistics, we derive the final refined statistics:

$$\mu_t \leftarrow \mu_t - \frac{\partial \mathcal{L}_{reg}}{\partial \mu_t} = \mu_t - 2\alpha_t(\mu_t - \mu_s) \tag{6}$$

$$\sigma_t \leftarrow \sigma_t - \frac{\partial \mathcal{L}_{reg}}{\partial \sigma_t} = \sigma_t - 2\alpha_t(\sigma_t - \sigma_s) \tag{7}$$

By integrating this discrepancy-aware statistical alignment and constraint mechanism with the class-balanced memory bank, our framework realizes unbiased online statistics estimation under realistic TTA settings.

### 4.4 Saliency-Guided Activation Sparsification and Gradient Reconstruction

Subsequently, the aligned running statistics are applied to normalize the test feature maps in a channel-wise fashion during test-time adaptation:

$$y_t(b,c,h,w) = \gamma_c \frac{x_t(b,c,h,w) - \mu_{t,c}}{\sqrt{\sigma^2_{t,c} + \epsilon}} + \beta_c \tag{8}$$

where $x_t \in \mathbb{R}^{\mathcal{B}\times C\times H\times W}$ denotes the input feature maps to each BN layer, $\mathcal{B}$, $C$, $H$, and $W$ denote the capacity, channels, height, and width of the feature map, respectively. The term $\epsilon$ is a small constant for numerical stability, and $\gamma_c$ and $\beta_c$ denote the learnable scale and bias parameters in BN layer.

However, fully caching $x_t$ for the input activations is often prohibitive for edge devices due to limited memory capacity. Consequently, selectively discarding the cache for channel $c$ of the activation $x_t$ facilitates a substantial reduction in memory usage. Nonetheless, such channel-wise sparsity inevitably disrupts the gradient flow during backpropagation, resulting in the ineffective update of affine parameters. To minimize memory overhead without compromising the effectiveness of parameter updates, we propose a saliency-guided activation sparsification and gradient reconstruction mechanism designed to facilitate the optimization process in Eq. 1.

Specifically, we prioritize caching channels with higher saliency scores while sparsifying less informative ones to maintain memory efficiency. We adopt the magnitude of the learnable scaling parameter, $|\gamma_c|$, as the proxy for channel saliency. This design is grounded in the intuition that for a linear combination $y = \sum w_i x_i$, the contribution of each channel is determined by the product of its weight and activation [Sun *et al.*, 2023]. In the context of normalization, Eq. 8 standardizes the feature maps of different channels to comparable scales. Consequently, the distinct scaling factors $\gamma_c$, learned independently for each channel, become the dominant indicators of channel significance to the subsequent

layers. Hence, we selectively sparsify the cached activations corresponding to the bottom $q\%$ of channels with the smallest $|\gamma_c|$ values in each BN layer to reduce storage overhead on edge devices.

While only a subset of activations is cached, the full-resolution output $y$ is still propagated to subsequent layers to ensure information continuity. In standard Batch Normalization, the gradient with respect to the input, $\nabla x$, is derived from the backpropagated output gradient $\nabla y$:

$$\begin{aligned} \hat{x} &= \frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}, \\ \nabla x &= \frac{\gamma}{\sqrt{\sigma^2+\epsilon}} \left(\nabla y - \mathbb{E}[\nabla y] - \hat{x} \cdot \mathbb{E}[\nabla y \cdot \hat{x}]\right). \end{aligned} \tag{9}$$

However, prior sparsification of $x_t$ along selected channels prevents the direct computation of $\nabla x$. To address this limitation, we propose an approximate gradient reconstruction operation. Specifically, for the $q\%$ sparsified channels, considering that the term $\frac{\gamma_c}{\sqrt{\sigma_c^2+\epsilon}}$ serves as the primary scaling factor, whereas the second and third terms arising from the batch mean and variance often exhibit severe fluctuations under small $\mathcal{B}$, consequently, we retain only the dominant direct gradient:

$$\nabla x_c \approx \frac{\gamma_c}{\sqrt{\sigma_c^2+\epsilon}} \cdot \nabla y_c \tag{10}$$

Consistent with the above approximation, $\nabla\gamma$ is derived from the inner product of $\nabla y$ and the normalized features of cached channels, whereas the gradient $\nabla\beta$ is computed by aggregating $\nabla y$. Detailed derivations and reconstruction specifics are provided in the Supplementary Material.

# 5 Experiments

## 5.1 Experiment Setting

To ensure a fair comparison, we strictly follow [Su *et al.*, 2024a], utilizing the identical realistic TTA protocol, network architectures, optimizers, and batch sizes.

**Datasets.** We evaluate all methods on three standard TTA benchmarks: CIFAR10-C, CIFAR100-C, and ImageNet-C [Hendrycks and Dietterich, 2019]. Each benchmark consists of 15 corruption types across five severity levels. Following standard protocols, we report results at the most challenging severity level (level 5). CIFAR10-C, CIFAR100-C, and ImageNet-C contain 10,000, 10,000, and 50,000 test samples per corruption, uniformly distributed over 10, 100, and 1,000 classes, respectively.

**Evaluation Metrics.** To evaluate both effectiveness and memory efficiency, we employ two metrics: **1) Classification Accuracy (%):** We report the classification accuracy under each corruption type at level 5 and the average accuracy across all corruption types. **2) Cache Size (MB):** We measure the total memory overhead by accounting for the intermediate activations required for backpropagation during adaptation, in comparison to which the cost of model parameters and optimizer states is negligible.

**Baselines.** We compare BOSA with parameter-free methods (Source, BN), conventional TTA approaches (TENT [Wang *et al.*, 2020], CoTTA [Wang *et al.*, 2022], PALM [Maharana *et al.*, 2025]), and methods designed for non-i.i.d. and realistic TTA settings (RoTTA [Yuan *et al.*, 2023], TRIBE [Su *et al.*, 2024a]). We further include recent memory-efficient TTA methods (EcoTTA [Song *et al.*, 2023], SURGEON [Ma *et al.*, 2025]), which primarily target the conventional TTA setting.

**Implementation Details.** For a fair comparison, we follow prior TTA studies [Yuan *et al.*, 2023; Su *et al.*, 2024a; Su *et al.*, 2024b] and adopt pre-trained WideResNet-28 [Zagoruyko and Komodakis, 2016] for CIFAR10-C and ResNeXt-29 [Xie *et al.*, 2017] for CIFAR100-C from the official RobustBench benchmark [Croce *et al.*, 2020]. For ImageNet-C, we use the standard pre-trained ResNet-50 [He *et al.*, 2016] provided by torchvision. Consistent with previous work [Su *et al.*, 2024a], we use a batch size of 64 and the Adam optimizer with a learning rate of 1e-3. In addition, we set $v_m$ to 0.999 for CIFAR10-C and CIFAR100-C, following [Yuan *et al.*, 2023], while adapting it to 0.99 for ImageNet-C. The sparsity factor $q$ is set to 0.7 for all datasets. The balanced memory bank adopts a design similar to that of [Yuan *et al.*, 2023], with a capacity of 64 for all datasets. All experiments are conducted on an NVIDIA Jetson TX2 edge platform.

## 5.2 Experiment Results

**Realistic Test-Time Adaptation Results.** Table 1 reports classification accuracy and cache usage on CIFAR10-C and CIFAR100-C under a realistic TTA setting with continuous adaptation across 15 corruption types. Several observations can be drawn from the results.

1) Conventional TTA methods, such as TENT, CoTTA, and PLAM, often fail to improve over the source model or simple BN adaptation, indicating that naive test-time optimization becomes unstable under realistic TTA settings, revealing the importance of designing more robust test-time adaptation methods.

2) Methods tailored for non-i.i.d. and realistic TTA, such as RoTTA and TRIBE, achieve substantial performance gains over conventional approaches. For example, on CIFAR10-C, TRIBE achieves a high average accuracy of 78.74%, slightly below BOSA's best performance of 79.05%. However, these gains come at the cost of excessive cache consumption. Adapting a batch of 64 samples, TRIBE requires over 1,182.79 MB of cache for backpropagation, primarily due to Balanced Batch Normalization and duplicated normalization layers across networks, which severely limits its practicality on resource-constrained devices.

3) Memory-efficient TTA approaches such as EcoTTA and SURGEON substantially reduce cache usage, but suffer from noticeably inferior adaptation performance, particularly under severe corruptions, highlighting the inherent trade-off between accuracy and memory efficiency. In contrast, BOSA consistently achieves strong performance on both CIFAR10-C and CIFAR100-C, while incurring significantly lower cache usage. These results demonstrate that BOSA effectively balances adaptation performance and memory efficiency, making it well-suited for realistic TTA on resource-constrained edge devices.

| Method | Mot. | Snow | Fog | Shot | Def. | Contr. | Zoom | Bright. | Frost | Elast. | Glas. | Gaus. | Pixel | JPEG | Imp. | Avg. Acc (%)↑ | Cache (MB)↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CIFAR10-C** | | | | | | | | | | | | | | | | | |
| Source | 63.94 | 74.82 | 74.32 | 34.17 | 52.21 | 52.28 | 58.98 | 90.42 | 57.92 | 72.89 | 44.82 | 27.86 | 40.75 | 68.06 | 26.69 | 56.01 | 0.00 |
| BN | 47.69 | 49.38 | 49.48 | 44.35 | 48.47 | 49.84 | 49.53 | 53.42 | 48.62 | 42.91 | 35.36 | 42.52 | 46.47 | 41.06 | 35.37 | 45.63 | 0.00 |
| TENT | 47.19 | 47.08 | 47.30 | 42.33 | 43.54 | 42.43 | 41.85 | 42.24 | 35.22 | 28.20 | 21.62 | 22.85 | 23.18 | 21.11 | 18.93 | 35.00 | 591.40 |
| CoTTA | 45.25 | 47.24 | 44.09 | 43.80 | 41.67 | 40.10 | 44.61 | 47.66 | 44.47 | 40.55 | 35.78 | 41.64 | 42.71 | 41.49 | 36.27 | 42.49 | 2480.73 |
| PALM | 47.68 | 49.16 | 48.94 | 45.53 | 47.77 | 46.13 | 49.42 | 51.51 | 46.61 | 40.73 | 34.12 | 38.44 | 36.69 | 33.74 | 27.53 | 42.93 | 1240.20 |
| RoTTA | 83.44 | 80.24 | 83.16 | 68.35 | 74.59 | 81.34 | 83.44 | 91.53 | 79.97 | 72.92 | **64.91** | 72.95 | 72.44 | 73.15 | 59.65 | 76.14 | 591.40 |
| TRIBE | 83.39 | 78.94 | **86.22** | **75.27** | 82.09 | 86.14 | 83.28 | 91.50 | **83.52** | **78.04** | 64.35 | **74.96** | 77.06 | 73.76 | 62.68 | 78.74 | 1182.79 |
| EcoTTA | 54.60 | 53.94 | 56.82 | 45.77 | 56.95 | 53.63 | 57.18 | 59.08 | 47.80 | 44.89 | 32.45 | 34.44 | 42.52 | 37.51 | 24.38 | 46.80 | 402.65 |
| SURGEON(BN) | 47.55 | 48.28 | 48.42 | 44.71 | 46.97 | 45.82 | 47.79 | 49.50 | 44.17 | 40.22 | 33.34 | 37.95 | 37.87 | 34.96 | 30.58 | 42.54 | 180.13 |
| BOSA | **84.25** | **81.74** | 85.82 | 70.81 | **82.88** | **86.65** | **87.28** | **91.60** | 81.30 | 76.28 | 64.87 | 74.91 | **78.65** | **75.34** | **63.36** | **79.05** | **169.00** |
| **CIFAR100-C** | | | | | | | | | | | | | | | | | |
| Source | 69.56 | 59.32 | 49.81 | 31.26 | 69.24 | 44.87 | 71.37 | 70.37 | 53.58 | 63.51 | 46.33 | 25.67 | 24.64 | 58.75 | 61.39 | 53.31 | 0.00 |
| BN | 63.68 | 57.24 | 52.00 | 52.19 | 63.19 | 60.25 | 64.91 | 65.64 | 58.11 | 58.42 | 52.15 | 51.15 | 58.15 | 50.99 | 50.90 | 57.27 | 0.00 |
| TENT | 64.30 | 55.83 | 48.27 | 33.72 | 31.58 | 18.21 | 15.63 | 10.65 | 6.82 | 5.03 | 4.18 | 4.04 | 4.06 | 3.73 | 3.12 | 20.61 | 914.36 |
| CoTTA | 63.23 | 57.38 | 52.17 | 54.85 | 61.88 | 58.30 | 64.97 | 64.39 | 58.70 | 58.69 | 55.00 | 54.81 | 61.83 | 55.69 | 53.82 | 58.38 | 3625.98 |
| PALM | 64.97 | 61.69 | 58.10 | **58.92** | 64.42 | 62.91 | 67.14 | 67.58 | 61.70 | 61.03 | 56.32 | 56.44 | 63.48 | 55.33 | 54.61 | 60.98 | 1812.73 |
| RoTTA | 65.68 | 57.68 | 54.01 | 50.34 | 63.76 | 59.65 | 64.53 | 72.50 | 60.78 | 62.61 | 57.05 | **57.45** | 62.12 | 58.84 | 58.11 | 60.34 | 914.36 |
| TRIBE | **71.33** | **65.19** | **61.40** | 58.67 | 68.05 | **69.89** | 72.86 | 73.29 | **67.45** | 64.48 | **58.71** | 57.02 | **64.59** | 58.79 | 59.58 | 64.75 | 1828.72 |
| EcoTTA | 63.53 | 55.57 | 53.13 | 51.07 | 62.93 | 61.25 | 66.30 | 67.85 | 58.05 | 58.60 | 44.89 | 49.59 | 57.89 | 54.78 | 54.14 | 57.30 | 654.31 |
| SURGEON (BN) | 64.95 | 59.51 | 56.38 | 54.00 | 59.30 | 57.44 | 58.50 | 60.51 | 53.14 | 51.75 | 45.98 | 42.89 | 47.20 | 37.54 | 33.73 | 52.19 | **211.96** |
| BOSA | 71.14 | 64.61 | 59.94 | 57.37 | **70.61** | 69.35 | **75.07** | **75.10** | 66.46 | **67.52** | 57.94 | 56.69 | 64.39 | **62.89** | **61.99** | **65.40** | 259.45 |
| **ImageNet-C** | | | | | | | | | | | | | | | | | |
| Source | 14.85 | 16.55 | 24.12 | 2.91 | 18.32 | 5.48 | 22.07 | 58.77 | 22.93 | 17.52 | 10.27 | 2.19 | 20.69 | 31.50 | 1.83 | 18.00 | 0.00 |
| BN | 26.36 | 33.93 | 47.19 | 15.51 | 14.95 | 17.34 | 38.04 | 63.96 | 31.40 | 43.56 | 15.15 | 14.69 | 47.71 | 39.23 | 14.95 | 30.93 | 0.00 |
| TENT | 35.63 | 40.27 | 48.80 | 22.53 | 18.30 | 11.28 | 17.62 | 23.09 | 6.36 | 4.57 | 1.20 | 1.02 | 1.61 | 1.10 | 0.60 | 15.60 | 2845.18 |
| PALM | 25.13 | 36.68 | 49.79 | 16.86 | 14.91 | 15.43 | 39.79 | **67.49** | 32.16 | 48.62 | 15.96 | 14.91 | 50.60 | 40.50 | 16.22 | 32.34 | 5574.75 |
| RoTTA | 25.14 | 29.98 | 44.74 | 14.45 | 14.63 | 21.39 | 39.00 | 65.69 | 35.35 | 47.17 | 23.84 | 14.57 | 51.30 | 47.59 | 21.63 | 33.10 | 2845.18 |
| TRIBE | 28.15 | 36.82 | 52.13 | 21.27 | 17.34 | 18.52 | **42.92** | 67.41 | **35.65** | **50.68** | 25.91 | 19.90 | 52.42 | 47.56 | 21.31 | 35.86 | 5690.36 |
| EcoTTA | 31.40 | 41.52 | 52.09 | 20.42 | 25.05 | **27.16** | 39.29 | 56.40 | 31.53 | 40.15 | 20.44 | 9.61 | 28.52 | 22.28 | 2.07 | 29.86 | 1335.89 |
| SURGEON (BN) | **38.18** | **43.48** | **52.59** | **27.09** | **25.15** | 26.79 | 37.00 | 50.09 | 30.37 | 39.94 | 23.93 | **22.87** | 34.43 | 31.82 | 22.53 | 33.75 | 1073.94 |
| BOSA | 27.15 | 34.23 | 48.29 | 21.60 | 19.65 | 23.93 | 42.39 | 66.43 | 35.12 | 49.29 | **28.64** | 22.38 | **52.84** | **49.35** | **27.23** | **36.57** | **807.89** |

Table 1: Classification accuracy (%) and cache usage (MB) for backpropagation under continuous adaptation to different corruptions at severity level 5 on CIFAR10-C, CIFAR100-C, and ImageNet-C under a realistic test-time adaptation setting. SURGEON (BN) denotes the BN-adapted variant of SURGEON as proposed in the original paper.

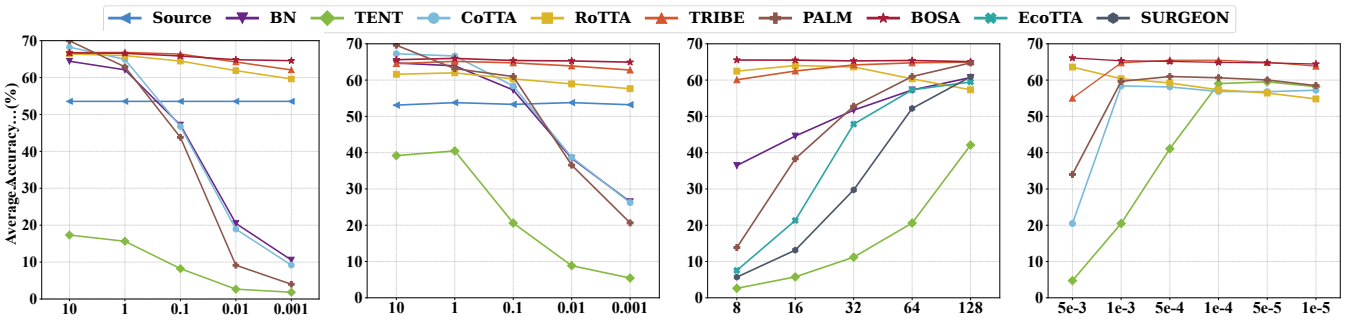


We further evaluate BOSA on the ImageNet-C dataset under a realistic TTA setting and report the classification accuracy and cache usage. As shown in table 1, BOSA outperforms existing methods in average accuracy while incurring the minimum memory overhead. These results demonstrate the effectiveness and memory efficiency of BOSA on large-scale datasets.

**Ablation Study.** To assess the contribution of each component in BOSA, we perform an ablation study on CIFAR10-C and CIFAR100-C in table 2. Removing the adaptive update constraint degrades performance, demonstrating its role in stabilizing statistics updates under dynamic distributions. Excluding online statistics alignment further reduces accuracy, underscoring its critical importance in mitigating biased statistics. Adapting without the balanced memory bank results in the most severe degradation, confirming the harm of directly learning from imbalanced test streams. Removing activation sparsification and gradient reconstruction preserves accuracy but incurs a substantial increase in cache usage. Finally, replacing the self-training with entropy minimization also reduces accuracy, showing that teacher-student consistency supervision provides more reliable guidance for continuous adaptation.

**Effect of Realistic TTA Streams Parameter $\rho$ and $\delta$.** Figure 4a shows that as $\delta$ decreases, BN, CoTTA, PALM, and TENT suffer sharp performance drops due to biased statistics under locally imbalanced streams. RoTTA and TRIBE also degrade markedly under severe local imbalance. In contrast, BOSA consistently achieves the highest and most stable accuracy across all $\delta$. Moreover, increasing the global imbalance factor $\rho$ from 1 to 10 (Figure 4a & 4b) causes negligible performance fluctuations for BOSA, demonstrating its robustness to both local and global imbalance through the class-balanced memory bank and the discrepancy-aware statistical alignment mechanism.

**Effect of Batch Size.** Considering that practical deployments often involve varying batch sizes, we evaluate performance across various batch sizes in Figure 4c. For a fair comparison, the memory bank capacities of both BOSA and RoTTA are set equal to the batch size. Results show that while most methods degrade significantly with smaller batches, BOSA and RoTTA remain relatively stable. Notably, RoTTA exhibits a performance drop at the smallest batch size of 8, whereas BOSA maintains optimal results across all settings, demonstrating its suitability for resource-constrained edge devices.




(a) $\delta$ when $\rho = 1$ (b) $\delta$ when $\rho = 10$ (c) Batch Size (d) Learning Rate

Figure 4: (a) & (b) Effectiveness of different methods under varying settings of $\delta$ and $\rho$ in the realistic TTA test streams. (c) Effectiveness of different methods under varying batchsize. (d) Sensitivity of different methods to the learning rate. All the above experiments are conducted on the CIFAR100-C dataset.

| Method | CIFAR10-C | | CIFAR100-C | |
|---|---|---|---|---|
| | Avg. Acc (%)↑ | Cache (MB)↓ | Avg. Acc (%)↑ | Cache (MB)↓ |
| **BOSA** | **79.05** | **169.00** | **65.40** | **259.45** |
| w/o adaptive update constraints | 75.27 | 169.00 | 59.70 | 259.45 |
| w/o Online Statistics Alignment | 69.68 | 169.00 | 58.56 | 259.45 |
| w/o Balanced Memory Bank | 65.07 | 169.00 | 57.78 | 259.45 |
| w/o Activation Sparsification | 78.64 | 564.00 | 65.12 | 872.00 |
| w/o Self-Training | 72.58 | 169.00 | 63.86 | 259.45 |

Table 2: Ablation study of different components of our BOSA on CIFAR10-C and CIFAR100-C. We report classification accuracy (%) and cache usage (MB) for metrics.

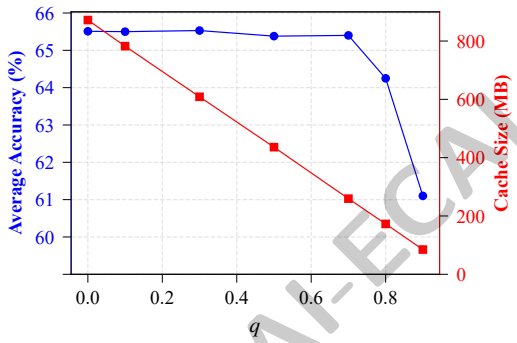



Figure 5: Accuracy–memory trade-off of BOSA under different sparsity ratios $q$, reporting both average accuracy and cache usage.

**Robustness of Learning Rates.** Figure 4d evaluates the sensitivity of different TTA methods to the learning rate under realistic TTA streams. Conventional methods such as TENT exhibit severe instability under suboptimal learning rates, while CoTTA and RoTTA improve robustness but still show noticeable performance fluctuations. In contrast, BOSA consistently maintains strong accuracy across a wide range of learning rates with only marginal variation, indicating reduced sensitivity to hyperparameter tuning and improved practicality for deployment in resource-constrained settings.

**Accuracy–Efficiency Trade-off.** Figure 5 illustrates the impact of $q$ on both average accuracy and the cache size. As $q$ increases linearly, the cache size decreases substantially, while BOSA demonstrates remarkable robustness to this sparsification. Specifically, increasing $q$ from 0.0 to 0.7 reduces memory consumption by over 60%, with less than a 1% drop in average accuracy, demonstrating a favorable accuracy–efficiency trade-off.

## 6 Conclusion

In this work, we tackle the practical challenge of enabling continuous model adaptation to imbalanced test streams on resource-constrained edge devices. To alleviate the inherent biases of imbalanced data, we propose a discrepancy-aware statistical alignment mechanism, coupled with a class-balanced memory bank to rectify statistics estimations under dynamic distributions. To reduce memory consumption, we propose a saliency-guided activation sparsification and gradient reconstruction mechanism, which selectively sparsifies activations based on channel saliency and reconstructs missing gradients for backpropagation. Moreover, a self-training strategy is integrated to promote stable adaptation. Extensive evaluations demonstrate that BOSA achieves an optimal trade-off between effectiveness and memory efficiency compared to existing TTA methods, underscoring its viability for real-world deployment on edge devices. We note that BOSA currently relies on normalization statistics, and extending the framework to architectures without normalization remains an interesting direction for future work.

## Acknowledgments

This work was supported in part by the Sichuan Provincial Science and Technology Achievement Transformation Project under Grant 2024ZHCG0009, the Natural Science Foundation of Sichuan, China under Grant 2026NSFSC1458, and the Jinjiang District 2025 "Open Bidding for Selecting the Best Candidates" Sci-Tech Project.

## References

[Adachi *et al.*, 2025] Kazuki Adachi, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Tomoki Hamagami. Test-time adaptation for regression by subspace alignment. 2025.

[Boudiaf *et al.*, 2022] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8334–8343, 2022.

[Brahma and Rai, 2023] Dhanajit Brahma and Piyush Rai. A probabilistic framework for lifelong test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3582–3591, 2023.

[Cai *et al.*, 2020] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33:11285–11297, 2020.

[Chi *et al.*, 2025] Zhixiang Chi, Li Gu, Huan Liu, Ziqiang Wang, Yanan Wu, Yang Wang, and Konstantinos N Plataniotis. Learning to adapt frozen clip for few-shot test-time domain adaptation. *arXiv preprint arXiv:2506.17307*, 2025.

[Croce *et al.*, 2020] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[Deng *et al.*, 2025] Qi Deng, Shuaicheng Niu, Ronghao Zhang, Yaofo Chen, Runhao Zeng, Jian Chen, and Xiping Hu. Learning to generate gradients for test-time adaptation via test-time training layers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16235–16243, 2025.

[Gong *et al.*, 2022] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

[Hakim *et al.*, 2023] Gustavo A. Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Ismail Ben Ayed, and Christian Desrosiers. Clust3: Information invariant test-time training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6136–6145, October 2023.

[Han *et al.*, 2025] Jisu Han, Jihee Park, Dongyoon Han, and Wonjun Hwang. When test-time adaptation meets self-supervised models. *arXiv preprint arXiv:2506.23529*, 2025.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Hendrycks and Dietterich, 2019] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[Hong *et al.*, 2023] Junyuan Hong, Lingjuan Lyu, Jiayu Zhou, and Michael Spranger. Mecta: Memory-economic continual test-time model adaptation. In *2023 International Conference on Learning Representations*, 2023.

[Hu *et al.*, 2025a] Youbing Hu, Yun Cheng, Zimu Zhou, Anqi Lu, Zhiqiang Cao, and Zhijun Li. Foctta: Low-memory continual test-time adaptation with focus. *arXiv preprint arXiv:2502.20677*, 2025.

[Hu *et al.*, 2025b] Zixuan Hu, Yichun Hu, Xiaotong Li, Shixiang Tang, and Ling-Yu Duan. Beyond entropy: Region confidence proxy for wild test-time adaptation. *arXiv preprint arXiv:2505.20704*, 2025.

[Jang and Chung, 2025] Minguk Jang and Hye Won Chung. Label distribution shift-aware prediction refinement for test-time adaptation. *Transactions on Machine Learning Research*, 2025:1–43, 2025.

[Jia *et al.*, 2024] Hong Jia, Young Kwon, Alessio Orsino, Ting Dang, Domenico Talia, and Cecilia Mascolo. Tinytta: Efficient test-time adaptation via early-exit ensembles on edge devices. *Advances in Neural Information Processing Systems*, 37:43274–43299, 2024.

[Lee *et al.*, 2024] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. *arXiv preprint arXiv:2403.07366*, 2024.

[Li *et al.*, 2025] Ziyue Li, Yang Li, and Tianyi Zhou. Skip a layer or loop it? test-time depth adaptation of pretrained llms. *arXiv preprint arXiv:2507.07996*, 2025.

[Liu *et al.*, 2025] Yating Liu, Xin Zheng, Yi Li, and Yanqing Guo. Test-time adaptation on recommender system with data-centric graph transformation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

[Ma *et al.*, 2025] Ke Ma, Jiaqi Tang, Bin Guo, Fan Dang, Sicong Liu, Zhui Zhu, Lei Wu, Cheng Fang, Ying-Cong Chen, Zhiwen Yu, et al. Surgeon: Memory-adaptive fully test-time adaptation via dynamic activation sparsity. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 30514–30523, 2025.

[Ma, 2024] Jing Ma. Improved self-training for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23701–23710, 2024.

[Maharana *et al.*, 2025] Sarthak Kumar Maharana, Baoming Zhang, and Yunhui Guo. Palm: Pushing adaptive learning rate mechanisms for continual test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19378–19386, 2025.

[Niu *et al.*, 2023] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

[Niu *et al.*, 2024] Shuaicheng Niu, Chunyan Miao, Guohao Chen, Pengcheng Wu, and Peilin Zhao. Test-time model adaptation with only forward passes. *arXiv preprint arXiv:2404.01650*, 2024.

[Niu *et al.*, 2025] Shuaicheng Niu, Guohao Chen, Deyu Chen, Yifan Zhang, Jiaxiang Wu, Zhiquan Wen, Yaofo Chen, Peilin Zhao, Chunyan Miao, and Mingkui Tan. Adapt in the wild: Test-time entropy minimization with sharpness and feature regularization. *arXiv preprint arXiv:2509.04977*, 2025.

[Sahoo *et al.*, 2024] Sabyasachi Sahoo, Mostafa Elaraby, Jonas Ngnawe, Yann Pequignot, Frédéric Precioso, and Christian Gagné. Layerwise early stopping for test time adaptation. 2024.

[Shin and Kim, 2024] Jin Shin and Hyun Kim. L-tta: Lightweight test-time adaptation using a versatile stem layer. *Advances in Neural Information Processing Systems*, 37:39325–39349, 2024.

[Song *et al.*, 2023] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023.

[Su *et al.*, 2022] Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. *Advances in Neural Information Processing Systems*, 35:17543–17555, 2022.

[Su *et al.*, 2024a] Yongyi Su, Xun Xu, and Kui Jia. Towards real-world test-time adaptation: Tri-net self-training with balanced normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15126–15135, 2024.

[Su *et al.*, 2024b] Zixian Su, Jingwei Guo, Kai Yao, Xi Yang, Qiufeng Wang, and Kaizhu Huang. Unraveling batch normalization for realistic test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15136–15144, 2024.

[Sun *et al.*, 2023] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

[Tan *et al.*, 2025] Mingkui Tan, Guohao Chen, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Peilin Zhao, and Shuaicheng Niu. Uncertainty-calibrated test-time model adaptation without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[Tsai *et al.*, 2024] Yun-Yun Tsai, Fu-Chen Chen, Albert Y. C. Chen, Junfeng Yang, Che-Chun Su, Min Sun, and Cheng-Hao Kuo. Gda: Generalized diffusion for robust test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23242–23251, June 2024.

[Wang *et al.*, 2020] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[Wang *et al.*, 2022] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[Wang *et al.*, 2023] Shuai Wang, Daoan Zhang, Zipei Yan, Jianguo Zhang, and Rui Li. Feature alignment and uniformity for test time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20050–20060, 2023.

[Wang *et al.*, 2024] Ziqiang Wang, Zhixiang Chi, Yanan Wu, Li Gu, Zhi Liu, Konstantinos Plataniotis, and Yang Wang. Distribution alignment for fully test-time adaptation with dynamic online data streams. In *European Conference on Computer Vision*, pages 332–349. Springer, 2024.

[Wang *et al.*, 2025] Zhijing Wang, Ji Guo, Xu Sun, Yi Luo, and Aiguo Chen. Double-boundary awareness of shared categories for source-free universal domain adaptation. *Neurocomputing*, page 131473, 2025.

[Xie *et al.*, 2017] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[Yang *et al.*, 2024] Xu Yang, Moqi Li, Jie Yin, Kun Wei, and Cheng Deng. Navigating continual test-time adaptation with symbiosis knowledge. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 5326–5334, 2024.

[Yuan *et al.*, 2023] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15922–15932, 2023.

[Zagoruyko and Komodakis, 2016] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[Zhao *et al.*, 2025] Shiji Zhao, Shao-Yuan Li, Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Dm-posa: enhancing open-world test-time adaptation with dual-mode matching and prompt-based open set adaptation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 7101–7109, 2025.