

On-demand Test-time Adaptation for Edge Devices

Xiao MA¹ Young D. Kwon² Dong MA¹

Abstract

Continual Test-time adaptation (CTTA) continuously adapts the deployed model *on every incoming batch of data*. While achieving optimal accuracy, existing CTTA approaches present poor real-world applicability on resource-constrained edge devices, due to the substantial memory overhead and energy consumption. In this work, we first introduce a novel paradigm – on-demand TTA – which triggers adaptation only when a *significant* domain shift is detected. Then, we present OD-TTA, an on-demand TTA framework for *accurate and efficient* model adaptation on edge devices. OD-TTA comprises three innovative techniques: 1) a lightweight domain shift detection mechanism to activate TTA only when it is needed, drastically reducing the overall computation overhead, 2) a source domain selection module that chooses an appropriate source model for adaptation, ensuring high and robust accuracy, 3) a decoupled Batch Normalization (BN) update scheme to enable memory-efficient adaptation with small batch sizes. Extensive experiments show that OD-TTA achieves comparable and even better performance while reducing the energy and computation overhead remarkably, making TTA a practical reality.

1. Introduction

Deep neural networks (DNNs) have achieved remarkable success in real-time edge tasks such as object detection (Wang et al., 2018), image recognition (Phan et al., 2020), and autonomous driving (Grigorescu et al., 2020). However, as a data-driven technique, DNNs typically *achieve optimal performance only when training and testing data share the same distribution* (Geirhos et al., 2018; Recht et al., 2019). In real-world scenarios, testing data often experiences distribution variations, known as *domain shifts*, due to factors such as weather changes, sensor noise, or light-

^{*}Equal contribution ¹Singapore Management University
²Samsung AI Center-Cambridge. Correspondence to: Dong MA <dongma@smu.edu.sg>.

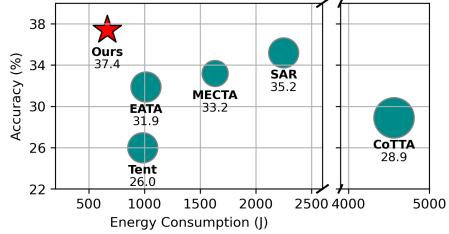


Figure 1. OD-TTA achieves a superior trade-off between memory, energy, and accuracy compared to state-of-the-art CTTA baselines. The radius of circles represents memory usage (See Appendix A.1).

ing conditions, which can result in significant performance degradation (Hendrycks & Dietterich, 2019).

To address this challenge, a plethora of previous works have developed **continual test-time adaptation (CTTA)** (Döbler et al., 2023; Wang et al., 2022; 2020; Niu et al., 2022; 2023; Hong et al., 2023; Zhang et al., 2022), which enables the pre-trained DNN model to continuously adapt to unseen domains using only unlabeled test data in either a self-supervised or unsupervised manner. Specifically, self-supervised learning approaches generate pseudo-labels for test data and fine-tune the model (Döbler et al., 2023; Wang et al., 2022). In contrast, unsupervised approaches, such as entropy minimization (Wang et al., 2020; Niu et al., 2022; 2023; Hong et al., 2023), are considered more efficient because they only update the model once per input batch using entropy loss. Recently, several approaches have focused on improving the efficiency of TTA to enable more practical deployment on resource-constrained devices. EATA (Niu et al., 2022) enhances efficiency by filtering out redundant data to reduce energy consumption during adaptation. EcoTTA (Song et al., 2023) and MECTA (Hong et al., 2023) aim to reduce memory overhead, and SAR (Niu et al., 2023) is designed to adapt the model with a batch size of 1. *However, existing efficient TTA approaches fail to fundamentally address the efficiency issue, as they still adhere to the CTTA paradigm which continuously executes resource-intensive backpropagation for each test batch.* Additionally, considering that the domain shift between consecutive batches is usually minor in real-world scenarios (Sun et al., 2022), CTTA may not yield substantial accuracy improvements.

In this paper, for the first time, we introduce a more practical and efficient paradigm, referred to as **on-demand TTA**.

Unlike continual TTA, on-demand TTA triggers model adaptation only when a significant domain shift that leads to an unacceptable (application-defined) performance drop occurs. This paradigm introduces several key challenges: (1) on-demand TTA requires continuous monitoring of the data distribution for every incoming sample/batch for potential domain shift detection. However, *efficiently* quantifying the domain shift (or performance drop) without labels is challenging and remains under-explored in existing TTA literature; (2) differing from continual TTA, where the distribution of consecutive batches usually remains similar, on-demand TTA inherently deals with more severe shifts after a domain shift is detected; (3) a notable limitation of existing Batch Normalization (BN)-based TTA is its dependence on *large batch sizes* (Wang et al., 2020; Niu et al., 2022), which requires considerable amount of memory.

To address the challenges, we propose **OD-TTA**, an end-to-end efficient **On-Demand TTA** framework designed for edge devices. We drew three key insights from our observations and experimental studies to guide the design of OD-TTA. First, we observed that entropy can be used not only for adaptation (as in existing methods (Wang et al., 2020)) but also for detecting domain shifts. Based on this, we devised a *novel lightweight domain shift detection mechanism* using exponential moving average (EMA) entropy to address the first challenge. Second, we found that adapting from different (similar or non-similar) source domains yields distinct post-adaptation performance. Therefore, instead of always adapting from the previous domain (as in continual TTA), we propose a *similar domain selection pipeline* that constructs and selects the closest domain for adaptation, resulting in better performance and faster convergence. Third, inspired by the insight that updating BN statistics and BN parameters consumes different amounts of memory and shows different sensitivity to batch sizes, we designed a *decoupled BN update scheme* that adapts the BN statistics and BN parameters asynchronously with different batch sizes, enabling effective model adaptation within a constrained memory budget.

We compare our proposed OD-TTA with strong baselines: Tent (Wang et al., 2020), CoTTA (Wang et al., 2022), EATA (Niu et al., 2022), SAR (Niu et al., 2023) and MECTA (Hong et al., 2023) on Cifar10-C (Hendrycks & Dietterich, 2019), ImageNet-C (Hendrycks & Dietterich, 2019), and SHIFT (Sun et al., 2022). Our proposed method achieves the best accuracy and energy efficiency over all the baselines while maintaining minimal memory requirements. Specifically, OD-TTA achieves up to 9.7% higher accuracy within comparable memory, and up to 47% energy saving on Cifar10-C. In particular, OD-TTA is the only effective method for BN-based models when operating with a batch size of 1. Note that other normalization layers such as Group Normalization compatible with a batch size of 1 perform

poorly with larger batch sizes as discussed in Section 4.2.

Our contributions are summarized as follows:

- We introduced the concept of on-demand TTA and presented OD-TTA, a novel on-demand TTA framework for edge devices. OD-TTA comprises a lightweight domain shift detector, a source domain selection module, and a decoupled BN updating strategy.
- We implemented OD-TTA on Jetson Orin Nano and evaluated its performance across multiple datasets. Our results indicate that OD-TTA achieves superior performance with minimal system overhead.
- Finally, our proposed paradigm and framework open the door, for the first time, to making TTA a practical reality with high accuracy and minimal system overheads on resource-constrained edge devices.

2. Related Work

Continual Test-Time Adaptation. We summarize existing CTTA methodologies into two categories: self-supervised and unsupervised learning paradigms.

Self-supervised CTTA initially generates pseudo labels for the testing data, then utilizes these labels to fine-tune the pre-trained model in a supervised manner. Wang et al. (Wang et al., 2022) developed a data augmentation method to construct a mean-teacher model for generating pseudo-labels. Bartler et al. (Bartler et al., 2022) introduced meta-learning to optimize the initial parameters of the model. Dobler et al. (Döbler et al., 2023) proposed the use of symmetric cross-entropy to replace the regular cross-entropy loss of the mean teacher adaptation, which proved to be better suited to the mean teacher approach.

Unsupervised CTTA addresses the domain shift by updating certain layers of the model through unsupervised loss functions. Benz et al. (Benz et al., 2021) highlighted the critical role of batch normalization (BN) layers in adapting to domain shifts. Following it, Wang et al. (Wang et al., 2020) proposed an early TTA method, TENT, which updates BN layers by simple entropy minimization. Then, Niu et al. (Niu et al., 2022) introduced EATA, which improves upon TENT by filtering out redundant and unreliable data and re-weighting remaining data. SAR (Niu et al., 2023) replaces the BN layer with a group normalization (GN) layer to make TTA work even when the batch size is one, but incurring high latency during adaptation. Moreover, recent works proposed memory-efficient adaptation such as MECTA (Hong et al., 2023) and Eco-TTA (Song et al., 2023). MECTA (Hong et al., 2023) adapted the BN layer to a novel MECTA normalization layer to reduce memory requirements, while Eco-TTA (Song et al., 2023) optimizes memory consumption during back-propagation by integrat-

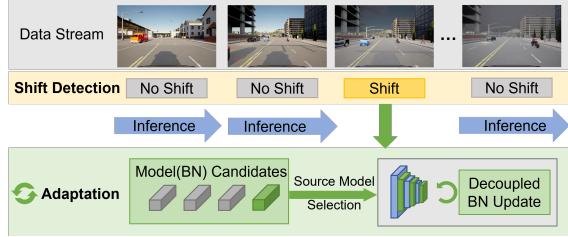


Figure 2. OD-TTA overview. The model performs regular inference while monitoring domain shifts. Once a shift is detected, OD-TTA selects the most similar BN candidate from a candidate pool and asynchronously adapts the BN statistics and affine parameters using a few new domain data.

ing lightweight meta-networks into the backbone. However, EcoTTA is not a straightforward plug-and-play method, as it requires redefining and retraining the model, while MECTA is easier to implement on existing pretrained models.

Our work differs from all existing research in that we proposed a completely new on-demand TTA paradigm and devised a suite of techniques to ensure it outperforms existing CTTA methods.

Domain Shift Detection. Domain shift detection is an essential part of OD-TTA, which monitors the distribution shift in the data stream to trigger the adaptation. Luo (Luo et al., 2022) and Vovk (Vovk et al., 2003) employs an auxiliary neural network to predict the martingale for each sample, serving as an indicator of domain shifts. Although this method effectively detects unpredictable domain shifts, it is memory-intensive because the auxiliary network (a dynamic CNN) must be continuously updated. Guy et al. (Bar Shalom et al., 2024) calculates a generalization bound for the source domain using the source dataset and identifies domain shifts by checking whether test samples exceed this established boundary. While this approach is less demanding in terms of memory usage during runtime, it is data-intensive, requiring substantial training data to establish an accurate generalization bound. Recently, Chakrabarty (Chakrabarty et al., 2023) and Niloy (Niloy et al., 2024) proposed using the mean of features extracted from a batch of data to represent the domain of the batch and reset the model to the source when the domain gap is over the threshold to achieve reliable CTTA. However, these feature-based methods rely heavily on large batch sizes, making them unsuitable for online data streams where data arrives sequentially and in smaller batches.

Our detection approach is both lightweight and effective, offering a significant advantage over other methods by being adaptable to any batch-size configuration.

3. On-demand Test-time Adaptation

3.1. Problem Formulation

In practical edge computing scenarios, sensor data arrive sequentially as $S_{\text{seq}} = \{s_1, s_2, \dots, s_t, \dots\}$, where s_t represents either a single sample or a small batch of samples arriving at time t . Domain shifts occur unpredictably, resulting in accuracy drop. On-demand TTA aims to adapt the model f_s only when a substantial domain shift results in unacceptable performance degradation. Following the CTTA setting, on-demand TTA operates under the constraint that the source dataset is not accessible during adaptation (Wang et al., 2020). Moreover, the adaptation must be performed directly on-device in an unsupervised manner, making it suitable for resource-constrained edge environments.

3.2. OD-TTA Overview

Figure 2 provides an overview of OD-TTA, which compromises two fundamental modules: domain shift detection and model adaptation. When a pre-trained model is deployed in real-world scenarios, it continuously performs inference on the incoming data stream while monitoring potential domain shifts using the proposed *lightweight shift detection mechanism*. Once a shift is detected, OD-TTA triggers an adaptation process involving two steps. First, OD-TTA *selects the closest domain from a pool of candidates (pre-trained or pre-adapted models)*, which can accelerate the subsequent adaptation process and enhance the adaptation performance by ensuring that adaptation starts from a more similar distribution. Second, OD-TTA adapts the model to align with the new domain data using a *decoupled BN updating strategy*, which effectively reduces the memory consumption while maintaining comparable accuracy.

3.3. Domain Shift Detection

The first objective in on-demand TTA is to detect the occurrence of domain shifts in a *lightweight* manner, as this needs to be performed continuously on all incoming data. However, since the ground truth labels of the test data are unavailable, monitoring accuracy drop caused by domain shift is challenging. Inspired by entropy minimization, which improves model performance by reducing the entropy of predictions *during training*, we draw the following insight:

Insight 1: During inference, the model accuracy is inversely correlated with the entropy of the predictions. The verification can be found in Appendix A.2.1.

This correlation arises because entropy measures the uncertainty in the model’s predictions (Wang, 2008). When there is a domain shift, the model tends to produce more uncertain predictions (higher entropy), as it struggles to generalize to the new distribution. This insight leads us to explore using

entropy as a potential metric to assess changes in model accuracy due to domain shifts.

EMA entropy calculation: However, certain samples may result in overly confident predictions (Wang et al., 2021), disrupting this inverse correlation. Considering that data arrives in a streaming manner during testing, sample-wise entropy cannot accurately characterize the model performance, as validated in Appendix A.2.1. Thus, we introduce an Exponential Moving Average (EMA) strategy to smooth the sample-wise entropy and incorporate historical entropy values, providing a more stable accuracy estimation. The formula for calculating the EMA entropy is as follows:

$$E_t = m \cdot E_{t-1} + (1 - m) \cdot x_t. \quad (1)$$

where E_t represents the EMA entropy at time t , m denotes the momentum factor (with a value between 0 and 1), and x_t is the entropy value of the current input sample at time t . The momentum m influences the stability of the EMA entropy and its sensitivity to domain shifts. A higher momentum value causes the current entropy to contribute minimally to EMA entropy, resulting in a more stable curve of EMA entropy values but reducing sensitivity to domain shifts.

Shift determination: After completing each adaptation process, OD-TTA records the EMA entropy over the next few samples (e.g., 100) as the entropy baseline (EMA_{base}), which reflects the current model’s capability on the adapted domain data. As OD-TTA calculates the sample-wise EMA entropy directly from the model inference outputs, referred to as EMA_{sample} , the extra computation is very lightweight. If $EMA_{sample} - EMA_{base}$ exceeds a user-defined threshold (EMA_{thr}), an adaptation is triggered. Setting the threshold (EMA_{thr}) is crucial for balancing sensitivity in shift detection and computational overhead in adaptation. We leave the impact of the threshold in Appendix A.2.2.

3.4. Source Domain Selection

CTTA always adapts the model from the *previous* domain, which may not be effective in on-demand TTA due to significant distribution shifts. Based on the key observation that different domains exhibit varying degrees of similarity (e.g., foggy and frost seem closer, compared to foggy and pixelate), we conducted experiments (Appendix A.3.1) and draw the insight below:

Insight 2: The source domain (i.e., the domain before each adaptation) can significantly impact the adaptation process, including convergence speed and post-adaptation accuracy.

This insight motivates us to select the domain most similar to the new domain from a candidate pool before adaptation, referred to as source domain selection. This process consists of two essential steps: (1) constructing an initial pool with

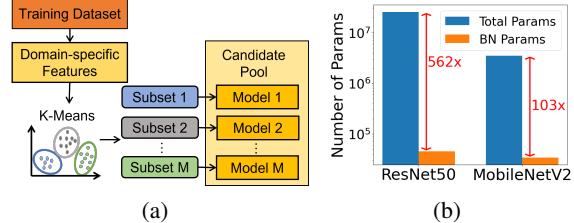


Figure 3. (a) Candidate pool construction; (b) storage comparison of saving only BN layers and the full model.

enough candidate models to ensure effectiveness from the outset, and (2) assessing domain similarity to identify the candidate most closely aligned with the new domain.

Candidate pool construction: Considering that a domain essentially reflects the distribution of a specific set of data, the process of creating a domain candidate pool can be converted as *generating multiple sets of data with diverse distributions*. Given that the training dataset naturally contains dispersed distributions due to the varied data sources and collection conditions, to construct the candidate pool, we propose to split the training dataset into multiple subsets and adapt the pre-trained model on each subset, as shown in Figure 3(a) (Details see Appendix A.3.2)¹.

Specifically, we first extract BN statistics that can represent the domain characteristics (i.e., domain features) for each training sample, given that BN statistics mainly capture data distribution (Niloy et al., 2024). Shallow BN layers are preferred because they capture more domain-specific features (See Appendix A.3.2 and Appendix A.3.3). Second, based on these domain features, we cluster the training samples into M subsets using the K-Means algorithm (MacQueen et al., 1967), with each cluster representing a domain². Third, we adapt the pre-trained model on each subset by only updating the BN layers in a supervised manner, resulting in M domain candidates in the pool by saving the BN layers. During runtime, we can further enlarge the pool by adding historical domains (pre-adapted models) that capture real-world domain characteristics.

As TTA usually only adapts the BN layer, we evaluate the storage overhead of saving only the BN layers compared to the entire model. For example, we find that for ResNet50, saving BN parameters (45.44K) accounts for only 1/562 of the full model (25.56M), indicating that even storing multiple domain candidates (e.g., 100) is negligible in terms of storage consumption, as shown in Figure 3(b).

Similar candidate selection: The second step involves selecting a candidate domain that is most similar to the new coming domain. The similarity measurement relies on extracting

¹Note that candidate construction relies solely on the pure training dataset, without any knowledge of the testing domains.

²Note that these domains (artificial) do not have physical meaning while simply representing different data distributions.

accurate domain features to represent the new domain using the source model. To address this issue, we cache N samples from the new domain (e.g., $N=128$), which are then processed through the source model to obtain the test BN statistics. Specifically, let the k -th batch of samples be denoted as $\{x_i\}_{i \in \mathcal{B}_k}$, where \mathcal{B}_k is the set of indices for the k -th batch. The BN mean for batch k , μ_k , is calculated as³:

$$\mu_k = \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \phi(x_i),$$

where $|\mathcal{B}_k|$ is the batch size, and $\phi(x_i)$ denotes the output of the second BN layer for sample x_i . (Verification see Appendix A.3.3.) The overall domain feature μ_{domain} is then computed as the average BN mean across all K batches:

$$\mu_{\text{domain}} = \frac{1}{K} \sum_{k=1}^K \mu_k.$$

Next, we compute the L2 distance between μ_{domain} , and the BN means μ_c of each candidate c in the pool. The candidate with the smallest distance is then selected as the most similar domain for subsequent adaptation:

$$c^* = \arg \min_c \|\mu_{\text{domain}} - \mu_c\|_2,$$

Here, c^* denotes the selected candidate that minimizes the L2 distance, ensuring that the closest match to the target domain is chosen for effective adaptation.

3.5. Decoupled BN Update

To achieve desirable performance, most existing CTTA approaches require large batch sizes (e.g., 64 in Tent and EATA), which consumes significant memory due to backpropagation. SAR (Niu et al., 2023) proposed to replace BN layers with Group Normalization (GN) layers to address the batch size issue. However, GN is inherently more computation-intensive and yields lower performance when batch size is large (Wu & He, 2018). We delved into the operation of BN layers with experiments (details see Appendix A.4) and derived the following insight:

Insight 3: Adapting only the BN layers with a small amount of data can achieve good performance. Updating BN statistics requires only a forward pass, which is memory-efficient yet highly sensitive to batch size. In contrast, updating BN parameters is less sensitive to batch size but involves backpropagation, which is more memory-intensive.

The observation motivates us to decouple the BN adaptation by updating of BN statistics in a larger batch size during inference and BN parameters in a small batch size with backpropagation, for the sake of memory saving.

³Capturing the BN statistics of the test batch requires only a single forward pass, incurring minimal memory overhead.

BN statistics update: Having cached N samples for similar model selection, we efficiently reuse these samples to form a small dataset to adapt the BN layers. Specifically, the samples are split into K batches. Inspired by the previous work (Yang et al., 2022), we employ an Exponential Moving Average (EMA) approach to integrate the BN statistics of the source model and the batches of new domain data, defined as follows:

$$S_k = (1 - \beta) \cdot S_{k-1} + \beta \cdot B_k, \quad (2)$$

where S_k represents the integrated BN statistics at batch k , β is the momentum factor, S_{k-1} is the BN statistics integrated from the previous batch, and B_k represents the BN statistics of the current batch. S_0 is the BN statistics of the selected candidate model. To ensure that the contributions from all batches are appropriately compiled with the source statistics, the momentum factor β is set to $1/K$.

BN parameters update: After updating the BN statistics, the selected model already captures the distribution of the new data, but the BN parameters still need to be fine-tuned accordingly through backpropagation. *To fit into the limited on-device memory, we aim to update the BN parameters with a small batch size (e.g., 1).* However, backpropagation using a single sample is challenging due to the inherent instability of unsupervised learning (Niu et al., 2023). To achieve stable fine-tuning, we introduce two strategies: (1) a sample filter to remove unreliable samples and (2) a contrastive loss as a regularization term to refine the entropy loss.

We define the overall loss for adaptation as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{entropy}} + \lambda \mathcal{L}_{\text{contrastive}},$$

where $\mathcal{L}_{\text{entropy}}$ is the regular entropy loss to ensure confident predictions. $\mathcal{L}_{\text{contrastive}}$ is a contrastive loss to regularize the adaptation process. λ is a weighting factor to balance the two components, empirically set to 0.05.

First, as noted in previous works (Niu et al., 2022; 2023), high-entropy samples can negatively impact entropy minimization, therefore we set a hard entropy threshold defined as $0.4 \cdot \log(C)$, following the methods proposed in EATA and SAR, where C is the number of classes and 0.4 is an empirically derived optimal constant.

Second, benefiting from the source model selection, we can obtain two distinct models: the poor source model (before candidate selection) and the current model (after updating BN statistics on the selected candidate). Inspired by contrastive learning (Jaiswal et al., 2020), the poor source model can be utilized as an anchor to guide the back-propagation process. This is achieved by constructing a contrastive loss as a regularization term alongside the entropy loss. Mathematically, let p represent the predictions from the current model undergoing adaptation, and p_{anchor} represent the predictions from the poor source model. The contrastive loss is then computed as:

$$L_{\text{contrastive}} = - \left(\frac{p - p_{\text{anchor}}}{\|p - p_{\text{anchor}}\|} \cdot p \right), \quad (3)$$

Minimizing the contrastive loss systematically drives the current model’s predictions away from those of the poor model and towards a trajectory that aligns more closely with the selected candidate model.

4. Evaluation

4.1. Experiment details

4.1.1. DATASETS

Cifar10-C and ImageNet-C (Hendrycks & Dietterich, 2019): a variant of the original CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009), designed for domain adaptation tasks. It is constructed by applying 15 different types of common corruption.

SHIFT (Sun et al., 2022): it is a domain shift dataset designed for autonomous driving systems that showcases three domain shifts including daytime → night, clear → foggy, and clear → rainy. Notably, the default resolution in SHIFT is 1280×800, which will lead to high latency and memory consumption on edge devices. To mitigate this issue, we follow the setting in (Sun et al., 2022) and reduce the image size to 640×400.

4.1.2. BASELINES

To the best of our knowledge, there are no existing works on on-demand TTA. Therefore, we compare OD-TTA with continual TTA baselines: CoTTA (Song et al., 2023), the self-supervised method; TENT (Wang et al., 2020), the first BN-based unsupervised method; EATA (Niu et al., 2022), a data-efficient BN-based approach; SAR (Niu et al., 2023), a Group-Normalization-based method for adapting with small batch sizes; and MECTA (Hong et al., 2023), a memory-efficient BN-based method. Notably, since MECTA is an extra component which can be added on existing BN-based baselines, we implement MECTA upon EATA by following the setting in the original paper.

4.1.3. ADAPTATION DETAILS

To ensure a fair comparison with baseline methods, we followed the optimal default settings outlined in the baseline papers. For our OD-TTA, we utilize 128/512 samples for the source domain selection and decoupled BN update on Cifar10-C/ ImageNet-C under batch size of 1. For batch size 16/64, we utilize 512 samples for adaptation for both datasets. In decoupled adaptation, we updated the BN statistics in batch size 16/256/256 and BN parameters in 1/16/64. (Memory usage see Figure 11(b).) The learning rates are set to 1×10^{-4} for CIFAR-10-C and 1×10^{-5} for ImageNet-C.

For the domain shift determination, we set the user-defined threshold EMA_{thr} at 0.06/0.3 for Cifar10-C/ImageNet-C, corresponding to an approximate accuracy drop of 5%. For the semantic segmentation task on the SHIFT dataset, we adapt the model using the learning rate of 1×10^{-4} and set the threshold EMA_{thr} at 0.1. In addition, we set a hard entropy threshold of 1.2 for CIFAR-10-C and 5.5 for ImageNet-C to trigger adaptation. This threshold ensures that adaptation will be activated when the model’s performance is critically low (below 10%).

4.1.4. IMPLEMENTATION

We evaluated OD-TTA on Jetson Orin Nano, a widely used edge device equipped with a Cortex-A78AE CPU and an NVIDIA Ampere GPU with 8GB RAM. For the software environment, we utilize Python 3.8 and PyTorch 2.0 on the Ubuntu 20.04 platform. Specifically, we evaluated the classification task with batch sizes of 1 and 16 on edge devices. For a batch size of 64 and the segmentation task, the evaluation was performed on a server, as the edge resources are too scarce to handle such evaluations.

4.2. Main Results

4.2.1. ACCURACY VS. MEMORY

Performance on Cifar10-C and ImageNet-C. We first evaluate the adaptation accuracy and memory consumption of OD-TTA using ResNet50 on CIFAR-10-C and ImageNet-C. The results in Table 1 demonstrate the effectiveness of our method in improving accuracy while maintaining low memory and energy consumption across various batch sizes on edge devices. Compared to state-of-the-art baselines, OD-TTA achieves a significant performance boost, particularly on CIFAR-10-C across all batch size settings. For ImageNet-C, OD-TTA outperforms other BN-based methods, including CoTTA, Tent, EATA, and MECTA, across all batch size settings. Notably, it stands out as the only BN-based approach capable of achieving high performance under a batch size of 1, which is critical for memory-constrained edge devices. While SAR, the GN-based baseline, can also handle batch size 1, it performs less effectively when operating with larger batch sizes. The superior performance of OD-TTA can be attributed to its effective source domain selection and the use of large batch sizes for updating BN statistics. The details are discussed in Appendix A.4.

In terms of memory consumption, our method demonstrates comparable GPU memory usage to existing TTA baselines such as Tent, EATA, and SAR under the same batch size. MECTA, on the other hand, is the most memory-efficient method, as it selectively updates BN layers and BN channels to minimize memory consumption. However, achieving comparable accuracy (e.g., 33.1% on ImageNet-C) requires MECTA to use 1231 MB, whereas OD-TTA achieves sim-

Table 1. Comparison of accuracy (%) on CIFAR-10-C and ImageNet-C using ResNet-50 along with memory consumption on Jetson Orin Nano.

Method	Batch Size = 1		Batch Size = 16		Batch Size = 64		
	Avg. Acc. (%)		Memory (MB)		Avg. Acc. (%)		
	Cifar10	ImageNet	Cifar10	ImageNet	Cifar10	ImageNet	(MB)
Source	59.5	26.9	242	59.5	26.9	358	59.5
CoTTA	10.0	0.1	889	78.1	28.9	3519	81.1
Tent	10.1	0.1	434	78.0	26.0	1728	81.0
EATA	22.8	0.8	506	78.3	31.9	1728	81.0
SAR	68.3	35.3	429	70.4	35.2	1723	69.2
MECTA	65.0	10.0	378	81.3	33.2	1231	81.3
Ours	78.0	33.1	414	83.0	37.4	1677	84.9
							40.4
							5763

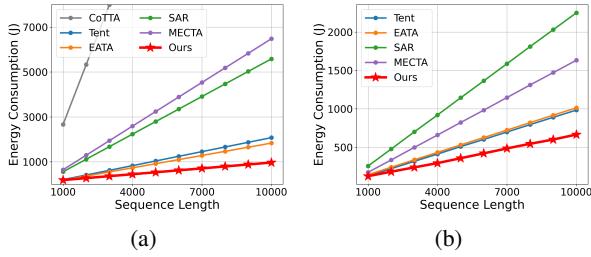


Figure 4. Energy consumption for processing domain data sequences of varying lengths under batch size = (a) 1 and (b) 16.

ilar accuracy with only 414 MB, highlighting its superior balance between memory efficiency and accuracy.

SAR and MECTA aim to address distinct challenges of the small batch size issue and memory consumption. However, both methods are less energy-efficient compared to other approaches. This will be further discussed in Figure 4.

Performance on SHIFT dataset. Given that semantic segmentation is significantly more computationally intensive than classification, we evaluate the performance on the SHIFT dataset using a batch size of 1. The results in Table 2 detail the mIoU scores across different domain shifts. We can observe that: (1) OD-TTA consistently outperforms all baselines across all types of domain shifts, achieving a significantly higher average mIoU; (2) the GN-based method, SAR, which outperforms BN-based baselines (CoTTA, Tent, EATA, and MECTA) in classification tasks, shows the poorest performance in segmentation tasks. The overall results highlight the robustness of OD-TTA and its ability to adapt effectively to real-world domain shifts, underscoring its practical utility in dynamic environments.

Energy Consumption. The energy consumption of on-demand TTA is closely related to domain shift frequency (the length of the domain data). In real-world scenarios, the frequency of domain shifts can vary significantly. When domain changes are infrequent or a single domain persists for an extended period, on-demand TTA achieves higher efficiency by minimizing the frequency of adaptations. To evaluate the energy efficiency of OD-TTA, we implemented all methods on the Jetson Orin Nano with batch sizes of 1

Table 2. Adaptation mIoU (%) on SHIFT along with memory consumption.

Method	SHIFT types			Avg.	Memory
	Day→Night	Clear→Foggy	Clear→Rainy		
Source	27.30	17.74	10.98	18.67	502
CoTTA	24.43	20.96	15.88	20.42	2065
Tent	24.72	22.12	17.71	21.52	1163
EATA	24.58	21.53	16.54	20.88	1216
SAR	23.61	7.63	4.22	11.82	1185
MECTA	24.23	20.59	15.42	20.08	855
Ours	31.08	25.17	19.05	25.10	1165

and 16 (64 is not available on the edge). The evaluation measured total energy consumption across domains of varying lengths, ranging from 1,000 samples (transient domains) to 10,000 samples (long-lasting domains).

Figure 4 illustrates the results, highlighting that OD-TTA achieves up to 47.1% energy savings compared to other adaptation methods in both batch size settings. Unlike CTTA methods which perform gradient-based updates for every batch of data, OD-TTA performs a one-time adaptation using only a few samples from the new domain. One-time adaptation eliminates the need for repeated computations during subsequent inference, resulting in consistently lower energy consumption. Notably, the energy-saving advantages of OD-TTA become increasingly significant as the domain persists for longer durations.

4.3. Detailed results

Then, we present the ablation study that assesses the effectiveness of different modules in OD-TTA.

4.3.1. ANALYSIS OF SHIFT DETECTION

To evaluate the domain shift detection module, we analyze its performance during the adaptation process on CIFAR-10-C. As shown in Figure 5, the EMA entropy fluctuates along the data stream, reflecting changes in domain characteristics and triggers when detecting an unpredictable increase. Results on more domain sequences are shown in Appendix A.5.3.

Untriggered shift. OD-TTA successfully detected 13 out of 15 domain shifts. The two undetected shifts occurred during transitions from Gaussian noise to shot noise and from motion blur to zoom blur. However, as reported in Table 7, these shifts did not result in accuracy drops. Specifically, the transitions from Gaussian noise to shot noise and motion blur to zoom blur actually led to accuracy improvements of 1.4% and 5.2%, respectively. These results demonstrate that OD-TTA avoids unnecessary adaptation when domain shifts do not substantially impact model performance.

Detection sensitivity. For the detected domain shifts, our

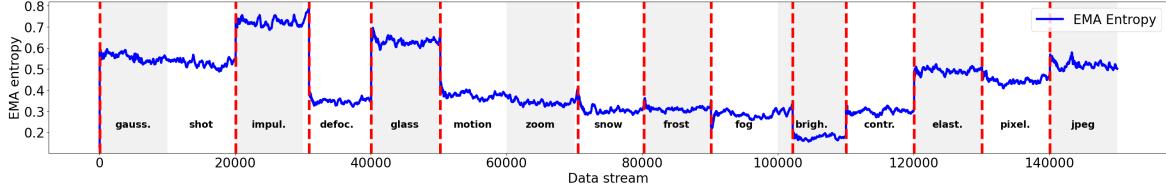


Figure 5. EMA entropy change along data stream on Cifar10-C. The red dotted lines are where domain shift is detected. Domains change after every 10,000 samples, as denoted by the changes in background color, which visually highlight transitions between domains.

Table 3. Evaluation of source model selection on CIFAR-10-C and ImageNet-C.

Adaptation from.	Batch Size = 1		Batch Size = 16		Batch Size = 64	
	Cifar10	ImageNet	Cifar10	ImageNet	Cifar10	ImageNet
Prev-Domain	79.5	27.8	84.4	31.4	86.0	33.4
Source-Model	80.6	32.8	84.6	35.4	85.9	37.5
Selected-Domain	81.0	34.6	85.7	36.7	86.1	39.1

method identified the shift within fewer than 100 samples for 7 domains, demonstrating its ability to quickly capture significant domain shifts. However, in the case of the transition from fog to brightness, it required 2168 samples from the new domain to detect the shift. This late determination is attributed to the relatively small accuracy drop of only 6.0% from fog to brightness, making the shift less pronounced and more challenging to detect promptly. However, it is unnecessary to adapt a model if the accuracy drop is small.

4.3.2. ANALYSIS OF SOURCE DOMAIN SELECTION

In Figure 3(a), we constructed a pool of domain candidates and selected the most similar candidate to initiate adaptation. Here, we further evaluate the source domain selection scheme. Specifically, we remove the domain shift detection module and directly adapt the model using decoupled adaptation with the first 1024 samples from each domain to ensure optimal adaptation performance. We compare it against two settings: adapting from the previous domain and adapting directly from the source model.

Table 3 shows the comparison results. The results demonstrate that, for every batch size setting and dataset, domain selection significantly enhances overall adaptation performance. Notably, adapting from the source domain outperforms continual adaptation but remains less effective than our source model selection approach.

4.3.3. ANALYSIS OF DECOUPLED BN UPDATE

In the decoupled BN update method, two primary factors influence adaptation performance: 1) the number of samples used for adaptation, and 2) the contrastive loss. In this section, we evaluate the impact of the two factors on ImageNet-C. Firstly, we explored the impact of the sample size on adaptation efficacy. We remove the detection module and adapt the model using the first few samples for

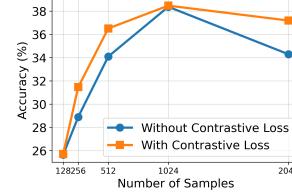


Figure 6. Impact of contrastive loss and number of samples.

Table 4. Adaptation accuracy (%) on CIFAR-10-C using MobileNetV2.

Method	BS=1	BS=16	BS=64
CoTTA	10.1	71.1	72.8
Tent	10.0	70.8	73.2
EATA	10.1	71.2	73.8
MECTA	12.4	11.3	14.4
Ours	68.3	73.8	74.4

adaptation. As shown in Figure 6, our findings indicate that increasing the number of adaptation samples enhances overall performance; however, the benefits become trivial when the number of samples exceeds 1024. Secondly, regarding the contrastive loss, the experimental results presented in Figure 6 reveal that employing this technique results in up to 2.6% accuracy improvement on the ImageNet-C, demonstrating its effectiveness in guiding the adaptation process towards the target domain. Additionally, we conduct a thorough comparison of adaptation under a few data with continual adaptation. We also evaluate the effectiveness of decoupled adaptation methods in detail in Appendix A.4.

4.3.4. EFFECTIVENESS ON MOBILENET

To evaluate the OD-TTA on other BN-based model architectures, we implement OD-TTA on MobileNetV2 (Sandler et al., 2018) on Cifar10-C. We did not compare OD-TTA with SAR, as GN/LN-based MobileNet is not publicly available. As Table 4 shows, OD-TTA consistently outperforms all the baselines in each batch size setting.

5. Conclusion

This paper proposes a novel concept called on-demand TTA, which triggers adaptation only when a domain shift is detected. We introduce OD-TTA, a framework designed to realize on-demand TTA for edge devices. OD-TTA comprises three key components: domain shift detection to monitor distribution shifts on the fly, source domain selection to optimize the efficacy of the source model for adaptation, and decoupled BN adaptation to update the model efficiently under limited memory constraints. The experiment result shows that OD-TTA significantly outperforms baselines while maintaining comparable memory overhead.

Impact Statement

In this work, we introduced a novel concept: on-demand test-time adaptation (TTA), a more practical and energy-efficient paradigm tailored for real-world edge devices. To realize this concept, we developed OD-TTA, an innovative on-demand TTA framework designed specifically for memory-constrained devices. OD-TTA enables efficient and selective adaptation, ensuring high performance while minimizing resource usage, making it a robust and scalable solution for real-world deployment. For the limitations of our OD-TTA, our approach is specifically designed for widely-used BN-based models. Other architectures, such as Vision Transformers, are not in the scope of this paper.

References

- Bar Shalom, G., Geifman, Y., and El-Yaniv, R. Window-based distribution shift detection for deep neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.
- Bartler, A., Bühler, A., Wiewel, F., Döbler, M., and Yang, B. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pp. 3080–3090. PMLR, 2022.
- Benz, P., Zhang, C., Karjauv, A., and Kweon, I. S. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 494–503, 2021.
- Chakrabarty, G., Sreenivas, M., and Biswas, S. A simple signal for domain shift. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3577–3584, 2023.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Döbler, M., Marsden, R. A., and Yang, B. Robust mean teacher for continual and gradual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7704–7714, 2023.
- Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018.
- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. A survey of deep learning techniques for autonomous driving. *Journal of field robotics*, 37(3):362–386, 2020.
- Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.
- Hong, J., Lyu, L., Zhou, J., and Spranger, M. Mecta: Memory-economic continual test-time model adaptation. In *2023 International Conference on Learning Representations*, 2023.
- Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.
- Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.
- Luo, R., Sinha, R., Sun, Y., Hindy, A., Zhao, S., Savarese, S., Schmerling, E., and Pavone, M. Online distribution shift detection via recency prediction. *arXiv preprint arXiv:2211.09916*, 2022.
- MacQueen, J. et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA, 1967.
- Niloy, F. F., Ahmed, S. M., Raychaudhuri, D. S., Oymak, S., and Roy-Chowdhury, A. K. Effective restoration of source knowledge in continual test time adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2091–2100, 2024.
- Niu, S., Wu, J., Zhang, Y., Chen, Y., Zheng, S., Zhao, P., and Tan, M. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.
- Niu, S., Wu, J., Zhang, Y., Wen, Z., Chen, Y., Zhao, P., and Tan, M. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.
- Phan, H., He, Y., Savvides, M., Shen, Z., et al. Mobi-net: A mobile binary network for image classification. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3453–3462, 2020.
- Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Song, J., Lee, J., Kweon, I. S., and Choi, S. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11920–11929, 2023.

Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., and Yu, F. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21371–21382, 2022.

Vovk, V., Nouretdinov, I., and Gammerman, A. Testing exchangeability on-line. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 768–775, 2003.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Wang, D.-B., Feng, L., and Zhang, M.-L. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021.

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

Wang, Q. A. Probability distribution and entropy as a measure of uncertainty. *Journal of Physics A: Mathematical and Theoretical*, 41(6):065004, 2008.

Wang, R. J., Li, X., and Ling, C. X. Pelee: A real-time object detection system on mobile devices. *Advances in neural information processing systems*, 31, 2018.

Wu, Y. and He, K. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

Yang, T., Zhou, S., Wang, Y., Lu, Y., and Zheng, N. Test-time batch normalization. *arXiv preprint arXiv:2205.10210*, 2022.

Zhang, M., Levine, S., and Finn, C. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022.

A. Appendix

A.1. Visualization of Accuracy, Memory and Energy

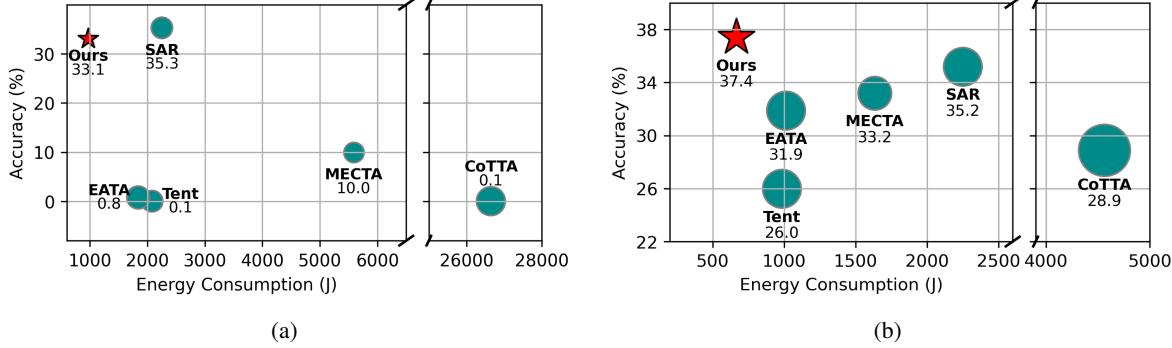


Figure 7. The trade-off of OD-TTA between memory usage, energy efficiency, and accuracy compared to state-of-the-art CTTA baselines on ImageNet-C under (a) batch size = 1 and (b) batch size = 16.

We visualized the comparison of OD-TTA with other baselines in terms of accuracy, memory, and energy consumption in Section 1. Specifically, we present results on ImageNet-C using ResNet-50 with a batch size of 16. To provide a clearer trade-off analysis, we include visualizations for both batch sizes of 1 and 16 in this section.

The energy consumption represents the total energy required to process 10,000 samples, as shown in Figure 4. Accuracy and memory consumption results are summarized in Table 1. Our results demonstrate that under both batch size settings, OD-TTA achieves high performance while consuming the least energy, all while maintaining comparable memory overhead. In contrast, SAR and MECTA, which are specifically designed to address the single-batch-size issue and memory constraints, respectively, fail to perform well in terms of energy efficiency.

A.2. More Explanation on Domain Shift Detection

In this section, we will give a thorough analysis of the domain shift detection mechanism.

A.2.1. CORRELATION BETWEEN ACCURACY AND ENTROPY

Inspired by entropy minimization in unsupervised learning, which improves model performance by reducing the entropy of predictions, we present the inverse correlation between model accuracy and average entropy as *Insight 1* in Section 3.3. This insight leads us to explore using entropy as a potential metric to assess changes in model accuracy due to domain shifts.

To verify *Insight 1*, we conducted an experiment using a pre-trained ResNet50 model on CIFAR-10-C. We first adapted the source model to four selected domains using supervised learning to ensure effective adaptation. Then, we tested the adapted models on several subsets⁴ sampled from other domains, calculating the average accuracy and entropy on each subset. The results are scattered in Figure 8(a), where each color represents one adapted model, and each scatter point denotes the accuracy and entropy of one subset. It is evident that entropy and accuracy are inversely correlated, with the relationship appearing nearly linear. This inverse correlation motivates us to approximate model performance by tracking the entropy of each input sample, without the need for ground-truth labels.

However, this correlation is demonstrated on a subset of samples (*subset-wise*) and does not consistently hold in real-world scenarios where data is processed in a streaming manner. In such cases, DNNs operate on each individual sample, resulting in significant variations in *sample-wise* entropy. Specifically, Figure 8(b) illustrates the sample-wise entropy (denoted by each scatter) when the streaming samples shift from the Source domain to the Impulse domain. It is evident that while most samples in the Impulse domain exhibit significantly higher entropy levels, some samples still present low entropy (i.e., outliers), hindering the direct detection of domain shifts using sample-wise entropy.

⁴From each domain, we constructed five subsets, resulting in a total of 75 subsets. Each subset consists of 500 randomly selected samples.

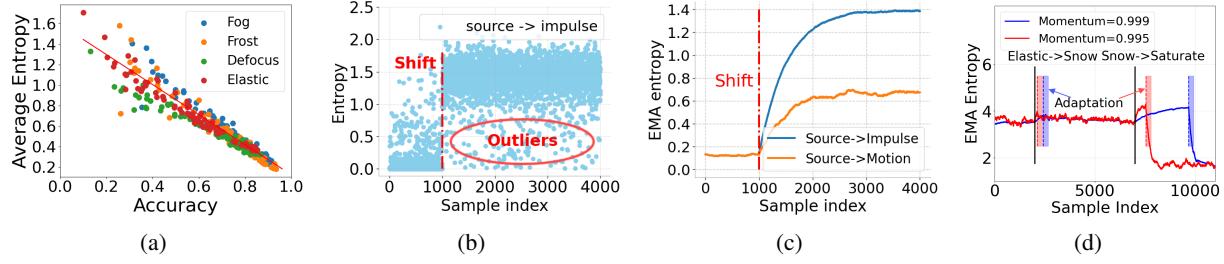


Figure 8. (a) The correlation between entropy and accuracy. (b) Sample-wise entropies cannot directly represent the accuracy because there are some over-confident data (outliers). (c) The EMA entropy will increase when there is a domain shift to different levels from source(96.1%) to motion blur (64.6%) and impulse noise (24.2%). (d) The impact of momentum in domain shift detection.

A.2.2. MORE EVALUATIONS OF EMA ENTROPY

We proposed the lightweight domain shift detection approach based on the EMA entropy. Figure 8(c) illustrates the EMA entropy curves for two adaptations, which clearly indicate the effectiveness of EMA entropy in detecting domain shifts.

A critical hyperparameter is the momentum value, denoted by m . To analyze the detection performance, we plot the EMA entropy changes across the domain sequence of ImageNet-C using ResNet50 under different momentum values in Figure 8(d). The x-axis represents the input sample stream, with domain changes marked by black vertical lines. The blue and red curves present the EMA entropy of the samples under the momentum of 0.999 and 0.995, respectively. Once a domain shift is detected, OD-TTA initiates adaptation for the next 256 samples, indicated by the shaded red and blue areas.

We can observe that: 1) both momentum settings (0.999 and 0.995) accurately capture domain shifts, promptly triggering the adaptation process, and 2) there is a noticeable trade-off between the EMA entropy fluctuation and detection sensitivity to domain shifts. While a momentum of 0.995 detects shifts more quickly, it also exhibits greater variance within a domain, potentially leading to unexpected triggers if the determined threshold is too low. In contrast, a momentum of 0.999 provides more robust detection, but requires more samples and time to confirm a shift; 3) following adaptation, the EMA entropy decreases to a new level, suggesting that each domain possesses a unique entropy signature.

A.3. Source Domain Selection

A.3.1. ADAPTATION FROM CLOSER DOMAIN RESULTS IN HIGHER EFFECTIVENESS

CTTA always adapts the model from the previous domain, which may not be effective in on-demand TTA due to significant distribution shifts. Based on the key observation that different domains exhibit varying degrees of similarity (e.g., foggy and frost seem closer, compared to foggy and sunny), we hypothesize that the source domain (i.e., the domain before adaptation) can significantly impact the adaptation performance. To test this, we adapted the model to the Snow domain (target domain) from three different source domains: Brightness, Saturate, and Gaussian noise. The accuracies of directly performing inference of the three source-domain models (i.e., models trained with the data from each domain only) on the Snow domain data were 85.2%, 74.1%, and 61.3%, respectively, which indicates that Brightness is the closest to Snow, followed by Gaussian noise and Saturate.



Figure 9. (a) Adaptation to a target domain from different source domains. (b) Clustering examples on the training set of ImageNet for candidate pool construction.

Figure 9(a) displays the adaptation accuracies with a different number of batches. We can observe that adapting from a closer domain (i.e., Brightness) yields higher accuracy after adaptation, suggesting that the source domain indeed affects the adaptation performance.

A.3.2. CANDIDATE POOL CONSTRUCTION

In candidate pool construction, a key challenge lies in extracting meaningful domain features for each sample in the training datasets. Specifically, we use the test sample BN mean from the second BN layer to represent the domain feature of each sample (the rationale for using second-layer BN mean as a domain characteristic is discussed in Appendix A.3.3).

From the previous work (Niu et al., 2023), we acknowledge that using testing batch BN fails to capture domain information effectively when the batch size is 1 (single sample). However, our findings indicate that this failure is primarily due to the inability of single samples to provide valid variance estimates. In contrast, the BN mean is less affected. Thus, we use the BN mean as the domain feature for each sample and cluster these features into subsets to create synthetic domains.

Figure 10 illustrates examples of clustering results. While we cannot explicitly define the domain of each subset, the clusters clearly exhibit distinct domain styles. For instance, cluster 1 (Top left in Figure 9(b)) predominantly contains samples with green, plant-like backgrounds, whereas cluster 5 (Bottom middle in Figure 9(b)) features samples in a white environment.

It is important to note that extracting domain features on a single sample is not an entirely accurate approach. However, it has proven to be an empirically effective method for clustering.

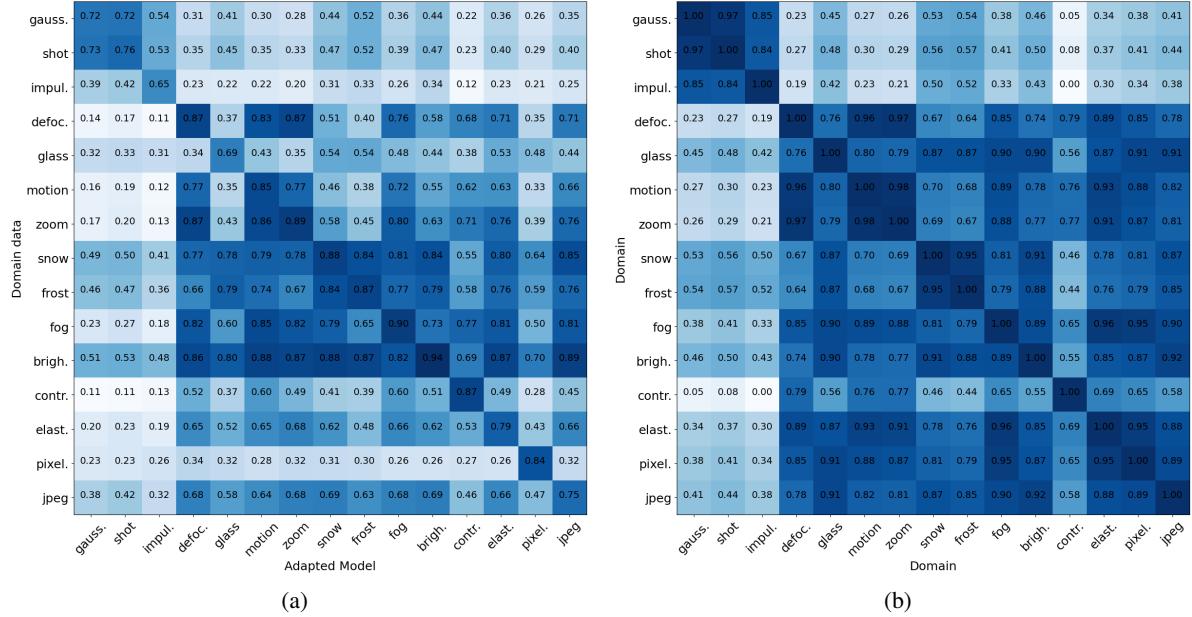


Figure 10. (a) Adapted model accuracy on other domains. (b) L2-based similarity calculated by BN mean from layer1.0.bn1 layer.

A.3.3. SIMILAR DOMAIN SELECTION.

In our work, we hypothesize that domain similarity can be effectively captured using the Batch Normalization (BN) statistics of models adapted to each domain. As discussed in Section 3.4, we utilized the BN statistics (mean) of the shallow BN layer to represent the domain information of a few batches of samples.

Defining domain similarity. Domains are considered similar if a model adapted to one domain performs well on another domain’s dataset. Conversely, if the model’s performance is poor, it suggests a low similarity between the domains. To quantify this, we use the performance similarity matrix as the ground truth, as shown in Figure 10(a). In detail, we first utilize the first 128 samples to adapt the source model to each target domain using EMA BN updates and entropy minimization, then we test the adapted model on all the domains in Cifar10-C. The performance of the domain A model on the domain B dataset can represent the similarity of the domains.

Measuring domain similarity via BN statistics. If the BN statistics of a model are adapted to a new domain, using a few

Table 5. Adaptation accuracy (%) of using a few samples from the domain (Few-data adaptation). We compared the results with continual Tent which uses entropy minimization on test batch statistics. To illustrate the key role of adapting the BN statistics, we also reported the results of only adapting the BN statistics using EMA BN without updating the affine parameters, as Few-data BN Stats. and adapting only through test batch statistics without updating affine parameters, as Continual BN Stats.

Method	gauss.	shot	impul.	defoc.	glass	motion	zoom	snow	frost	fog	brigh.	contr.	elast.	pixel.	jpeg	Avg. Acc
Continual BN Stats.	68.2	71.3	60.3	83.5	64.6	81.4	85.2	84.6	83.5	86.8	90.7	84.7	74.5	78.8	70.7	77.9
Continual Tent	68.8	72.8	62.5	83.2	65.3	81.3	84.8	83.6	83.3	85.2	89.6	84.8	75.8	79.2	72.3	78.2
Few-data BN Stats.	67.9	73.7	62.0	72.1	65.8	79.4	88.6	88.4	85.4	88.5	92.9	80.8	76.9	70.3	71.7	77.6
Few-data Adapt.	67.9	73.7	62.3	72.2	66.1	79.4	88.6	88.5	85.4	88.6	92.9	81.4	76.9	71.0	72.1	77.8

samples, it can be utilized as a representation of the domain. To evaluate the effectiveness, we compare the running mean values from the Batch Normalization (BN) layers of models adapted to these domains. For any two domains i and j , the similarity is calculated using the L2 distance between their BN statistics:

$$d_{ij} = \|\mu_i - \mu_j\|_2,$$

where μ_i and μ_j are the running mean vectors extracted from a specific BN layer of models adapted to domains i and j , respectively.

To normalize these distances into a similarity score in the range $[0, 1]$, we apply the following transformation:

$$s_{ij} = 1 - \frac{d_{ij} - d_{\min}}{d_{\max} - d_{\min}},$$

where d_{\min} and d_{\max} are the minimum and maximum distances across all domain pairs. The resulting similarity matrix S captures how closely each pair of domains is related based on their BN statistics. As Figure 10(b) shows, the similarity calculated using the second BN layer can be closely aligned with the accuracy matrix in Figure 10(a), which is the ground truth of the similarity.

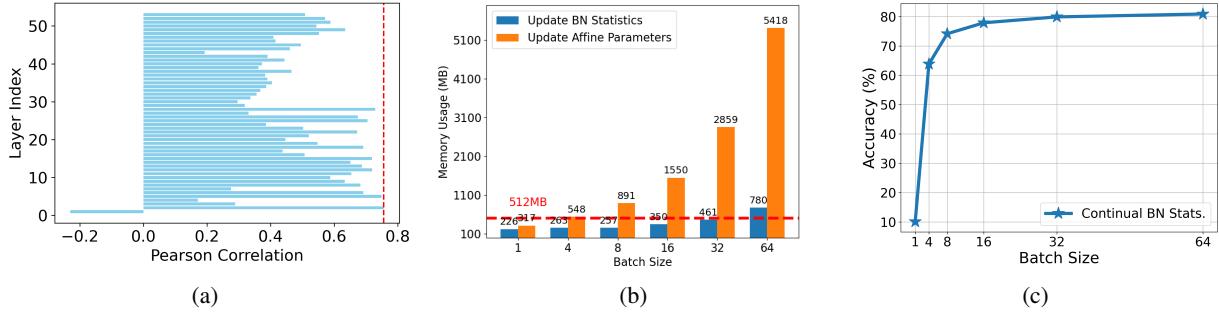


Figure 11. (a) BN layer correlation to the true domain similarity. The layers are ordered from shallow to deep, and their correlations are represented as a bar chart. (b) The memory consumption of updating BN statistics (forward pass) and affine parameters (backward pass). The memory usage is measured on the server.

Which BN layer provides the best measurement? By leveraging the similarity matrix derived from BN statistics, we aim to assess its alignment with the accuracy matrix, which serves as the ground truth. This alignment is quantified using the Pearson correlation coefficient between the similarity matrices of the BN statistics and the performance.

As shown in Figure 11(a), the second BN layer demonstrates the strongest alignment with the ground truth. This indicates that it can serve as a reliable metric for measuring domain similarity.

Table 6. Adaptation accuracy (%) of decoupled BN adaptation (Decouple Adapt.) compared to conventional synchronized adaptation using 128 samples. We compared the decoupled adaptation with three baselines: (1) BN Stats: adapting only the BN statistics; (2) BN Adapt: jointly updating BN statistics and affine parameters through entropy minimization; (3) Adapt. with Filter: further adapting the model using only low-entropy data.

Method	gauss.	shot	impul.	defoc.	glass	motion	zoom	snow	frost	fog	brigh.	contr.	elast.	pixel.	jpeg	Avg. Acc
BN Stats.	67.9	73.7	62.0	72.1	65.8	79.4	88.6	88.4	85.4	88.5	92.9	80.8	76.9	70.3	71.7	77.6
BN Adapt.	67.9	73.7	62.3	72.2	66.1	79.4	88.6	88.5	85.4	88.6	92.9	81.4	76.9	71.0	72.1	77.8
BN Adapt. with Filter	68.9	74.6	61.7	70.6	64.8	79.8	88.5	87.8	87.0	87.6	92.5	82.5	77.3	71.4	72.2	77.8
Decoupled Adapt.	70.2	75.1	62.7	74.7	66.5	81.6	89.7	89.6	87.1	89.9	94.1	83.8	78.5	76.0	73.3	79.5

A.4. Decoupled BN Adaptation

A.4.1. ADAPTION FROM ONLY A FEW DATA CAN MITIGATE DOMAIN SHIFT

Inspired by previous work (Benz et al., 2021), corruption robustness can be improved by capturing the BN statistics of only 32 samples. We hypothesize that adapting to a domain using a small number of samples is sufficient to achieve a robust model for the remaining samples in that domain. By employing the EMA strategy to update BN statistics and entropy minimization to update the affine parameters on 128 samples in CIFAR-10-C, the inference accuracy on the remaining samples is illustrated in Table 5. We compared the few-data adaptation with Tent, the continual adaptation method based on test batch statistics and entropy minimization. The momentum of few-data adaptation is set to 0.15.

In Table 5, Continual BN Stats. continuously updates the Batch Normalization (BN) statistics using test batch statistics. Continual Tent adapts the BN affine parameters through entropy minimization while also updating the BN statistics using test batch statistics. Few-data BN Stats. leverages 128 samples to update the BN statistics via an Exponential Moving Average (EMA) approach. Few-data Adapt. further extends this by utilizing the 128 samples to update both the BN statistics with EMA BN and the affine parameters through entropy minimization.

We can see that the few-data adaptation achieves comparable results to continual Tent across most corruption types, with a minimal difference in the average accuracy (77.8% vs. 78.2%). This demonstrates that adapting to the domain using a small subset of samples is sufficient to maintain robust performance on the remaining data. Moreover, this approach enables adaptive triggering of adaptation using only a few samples, making it both practical and efficient.

A.4.2. ADAPTING BN STATISTICS VS. AFFINE PARAMETERS

The results of Continual BN Stats. and Continual Tent in Table 5 illustrated that updating the BatchNorm (BN) statistics is crucial for improving adaptation performance, as demonstrated by Figure 11(c). The sensitivity of BN statistics to batch size is evident from the performance degradation with smaller batch sizes. This indicates that a sufficient number of samples per batch is necessary to capture reliable BN statistics for effective adaptation. However, as shown in Figure 11(b), updating BN statistics with larger batch sizes is feasible from a memory perspective. Even with a batch size of 32, the memory usage for updating BN statistics is significantly lower than updating the affine parameters with a batch size as small as 4. Moreover, with a batch size of 4, the memory requirement for updating BN parameters of ResNet50 can easily exceed the available memory space of some edge devices such as the Raspberry Pi Zero 2W with only 512MB DRAM.

This comparison highlights the efficiency of adapting BN statistics alone. It not only delivers robust performance when sufficient batch size is used but also enables memory-efficient adaptation. This balance between performance and resource usage makes updating BN statistics a practical and effective approach for domain adaptation, especially when resources are constrained.

A.4.3. DECOUPLED ADAPTATION EVALUATION

After confirming that using a small amount of data for adaptation can achieve performance comparable to continual adaptation, we further evaluate the decoupled BN adaptation strategy. The Few-data Decoupled Adaptation results in Table 6 demonstrate the effectiveness of this approach, where BN statistics are updated using a batch size of 16, and affine parameters are adapted with a batch size of 1. We compared the decoupled adaptation with three baselines in a batch size

Table 7. Comparisons with state-of-the-art methods on Cifar10-C (severity level 5) under batch size of 1 regarding Accuracy (%). The bold number indicates the best result.

Method	Noise			Blur				Weather				Digital			Avg.	
	gauss.	shot	impul.	defoc.	glass	motion	zoom	snow	frost	fog	brigh.	contr.	elast.	pixel.	jpeg	
Source	31.9	38.4	24.2	68.8	45.0	64.6	74.3	83.9	76.7	80.1	91.3	45.2	69.1	28.0	71.2	59.5
CoTTA	10.2	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0
Tent	10.9	9.9	10.2	10.0	9.9	10.2	10.0	9.9	10.1	10.1	9.9	10.0	10.1	9.9	9.9	10.1
EATA	17.5	18.7	13.9	22.8	15.8	23.2	23.6	27.7	26.0	31.2	32.0	32.8	18.5	22.1	17.0	22.8
SAR	54.2	58.6	41.9	79.6	41.9	75.3	82.0	83.4	79.4	79.2	91.9	87.6	68.3	27.7	73.1	68.3
MECTA	54.6	57.0	48.2	69.3	47.3	67.7	70.3	74.1	71.3	75.9	79.4	81.2	58.7	64.6	55.5	65.0
Ours	71.5	72.9	67.9	79.0	47.8	80.9	86.1	87.1	87.0	89.2	83.2	86.8	78.8	76.0	75.4	78.0

Table 8. Comparisons with state-of-the-art methods on ImageNet-C (severity level 5) under batch size of 1 regarding Accuracy (%). The bold number indicates the best result.

Method	Noise			Blur				Weather				Digital			Avg.	
	gauss.	shot	impul.	defoc.	glass	motion	zoom	snow	frost	fog	brigh.	contr.	elast.	pixel.	jpeg	
Source	18.1	20.1	17.4	19.7	10	22.2	26.5	32.3	32.9	38.7	68.3	25.3	14.1	12.8	44.7	26.9
CoTTA	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.1
Tent	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	0.2	0.2	0.1
EATA	0.3	0.3	0.3	0.3	0.1	0.4	0.7	1.1	1.4	1.7	2.2	1.0	0.7	0.8	0.7	0.8
SAR	20.4	27.8	31.2	20.0	17.4	25.3	27.6	42.1	44.9	45.6	67.5	40.4	23.5	41.5	54.1	35.3
MECTA	6.9	8.7	7.7	5.7	5.8	8.8	11.1	16.6	12.7	18.7	22.7	0.2	8.8	9.0	7.4	10.0
Ours	28.2	21.8	28.1	10.9	10.1	13.6	41.3	43.2	38.7	57.3	72.3	11.5	42.3	28.1	49.9	33.1

of 16: (1) BN Stats.: adapting only the BN statistics; (2) BN Adapt.: jointly updating BN statistics and affine parameters through entropy minimization; (3) Adapt. with Filter: further adapting the model using only low-entropy data following EATA and SAR. The results indicate that decoupled adaptation not only outperform the few-data adaptation by 1.7%, but also perform better than continual Tent in Table 5. The improvement is attributed to updating BN statistics before entropy minimization, which helps guide the entropy minimization step in a more accurate direction during unsupervised adaptation.

In all baseline methods, we use a batch size of 16. For our Decoupled Adaptation approach, we employ a batch size of 16 for computing BN statistics while using a batch size of 1 for updating affine parameters. As a result, our memory consumption is equivalent to that of BN Adaptation with a batch size of 1. The results indicate that our method achieves higher performance while maintaining low memory overhead.

A.5. More Evaluation on Cifar10-C and ImageNet-C.

A.5.1. DETAILS OF THE MAIN RESULTS

The additional results presented in the appendix (Table 7 and Table 8) provide a granular view of OD-TTA’s performance across various corruption types and under a batch size of 1.

On CIFAR-10-C (Table 7), OD-TTA demonstrates a significant performance advantage over state-of-the-art methods across almost all corruption types under severity level 5. While SAR achieves the highest accuracy in specific corruptions, such as defocus blur, brightness, and contrast, OD-TTA outperforms the baselines overall, achieving an average accuracy of 78.0%. This marks a significant improvement over SAR (68.3%) and MECTA (65.0%), underscoring OD-TTA’s effectiveness in adapting to diverse corruptions while maintaining competitive memory efficiency.

For ImageNet-C (Table 8), OD-TTA maintains competitive performance and strikes a balance between accuracy and memory consumption. While SAR achieves slightly higher average accuracy (35.3%) due to strong performance in specific noise and blur categories, OD-TTA performs better across a wider range of corruptions, particularly in weather-related corruptions

like fog (57.3%) and digital corruptions like JPEG compression (49.9%). With an average accuracy of 33.1%, OD-TTA demonstrates robustness in handling domain shifts, even in large-scale and complex datasets like ImageNet-C.

The results further emphasize the advantages of OD-TTA. First, OD-TTA demonstrates consistent performance across corruption types. While some baselines perform well in specific categories, OD-TTA achieves high accuracy consistently across all corruption types, highlighting its generalizability. Second, despite achieving near state-of-the-art accuracy, OD-TTA maintains a significantly lower memory footprint compared to other methods, particularly Tent and CoTTA. Third, the ability to handle a batch size of 1 with high accuracy (78.0% on CIFAR-10-C, 33.1% on ImageNet-C) sets OD-TTA apart as a practical solution for edge scenarios.

A.5.2. DOMAIN SHIFT DETECTION ON IMAGENET-C

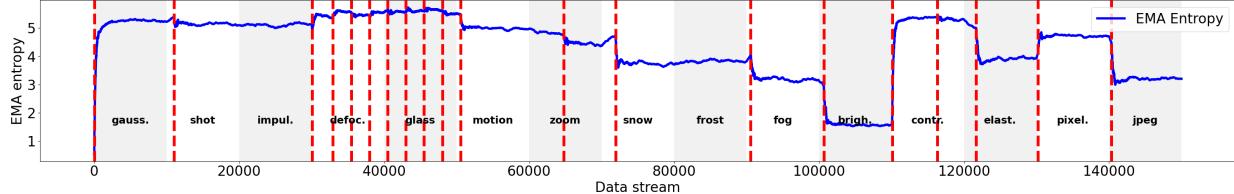


Figure 12. EMA entropy change along data stream on ImageNet-C. The red dotted lines are where domain shift is detected. Domains change after every 10,000 samples, as denoted by the changes in the background color.

We also evaluate the detection performance on ImageNet-C. As shown in Figure 12, the EMA entropy fluctuates along the data stream, reflecting changes in domain characteristics and triggers when detecting an unpredictable increase.

Untriggered shift. OD-TTA successfully detected 13 out of 15 domain shifts. The undetected shifts occurred during transitions from shot noise to impulse noise and from snow to frost. The untriggered detection did not incur significant accuracy drop. Specifically, the transitions from shot noise and motion blur to impulse led to accuracy improvements of 21.8%. The shift from snow to frost only results in 4.5% performance drop. These results demonstrate that OD-TTA avoids unnecessary adaptation when domain shifts do not substantially impact model performance.

False Trigger. Compared to the CIFAR-10-C results in Figure 5, the ImageNet-C results exhibit 7 unnecessary false triggers, particularly in defocus blur, glass blur, and contrast. These false detections can be attributed to the model’s poor performance even after adaptation. Notably, the accuracy remains low at 10.9%, 10.1%, and 11.5% for these corruptions, indicating that the adaptation process struggles to improve performance. Since we set a hard threshold to trigger adaptation when accuracy drops below approximately 10%, frequent triggers occur in these poorly performing domains. As a result, the detection scheme attempts multiple adaptation steps in an effort to optimize the model, leading to unnecessary re-triggering.

A.5.3. IMPACT OF DOMAIN CHANGE ORDER

The order of domain changes can significantly influence the performance of OD-TTA, as the domain shift detection varies in different domain orders. Table 9 reports the adaptation accuracy (%) of different methods on a randomly generated domain change sequence for CIFAR-10-C and ImageNet-C.

On CIFAR-10-C, OD-TTA achieves the highest average accuracy of 79.5%, demonstrating robust adaptation across diverse domain shifts. Compared to SAR (68.2%) and MECTA (63.4%), OD-TTA consistently outperforms the baselines, especially in challenging corruptions such as frost, fog, and snow. The strong performance of OD-TTA across varying domains highlights its ability to adapt effectively regardless of the sequence of domain changes.

For ImageNet-C, OD-TTA achieves an average accuracy of 32.0%, outperforming Tent and MECTA, and closely competing with SAR which achieves 37.4%. Notably, OD-TTA shows significant improvements in corruptions like fog (55.9%) and brightness (72.7%), which are particularly challenging for other baselines.

The domain order impacts the domain shift detection as different types of two adjacent fields will result in different accuracy changes. Figure 13 illustrates the EMA entropy changes along the data stream in the random domain change sequence. In this order, the domain shift detection mechanism effectively captures all domain shifts. On average, it requires 510 samples to detect a domain shift. Specifically, 9 out of 15 shifts are detected within 300 samples, demonstrating the efficiency of the approach. However, one shift—from JPEG to motion blur—is detected more slowly, requiring 2,941 samples. It is attributed

Table 9. Adaptation accuracy (%) on a random domain change sequence on Cifar10-C and ImageNet-C.

Dataset	Method	jpeg	motion	frost	contra.	zoom	gauss.	fog	defocus	snow	shot	bright.	pixel.	elast.	impul.	glass.	Avg. Acc
Cifar10-C	Source	71.2	64.6	76.7	45.2	74.3	31.9	80.1	68.8	83.9	38.4	91.3	28.0	69.1	24.2	45.0	59.5
	CoTTA	10.1	10.1	10.0	10.2	10.0	10.0	9.9	10.0	10.1	9.8	10.2	10.1	10.1	10.2	10.1	10.1
	Tent	10.9	10.1	10.2	10.9	10.0	10.0	10.9	10.0	10.9	9.8	10.6	10.1	10.0	10.4	10.1	10.3
	EATA	17.0	23.2	26.0	32.8	23.6	17.5	31.2	22.8	27.7	18.7	32.0	22.1	18.5	13.9	15.8	22.8
	SAR	73.5	75.9	79.7	87.7	82.0	52.5	79.0	79.6	83.6	58.8	91.9	27.8	68.4	43.4	39.6	68.2
	MECTA	57.9	71.0	73.4	20.5	73.6	57.0	77.7	72.2	76.2	59.4	82.6	66.1	62.4	49.9	50.8	63.4
	Ours	75.6	80.1	85.7	85.4	88.5	71.3	88.3	85.7	87.9	74.2	93.1	75.4	77.0	59.4	65.3	79.5
ImageNet-C	Source	45.4	21.6	33.3	25.4	26.0	18.1	39.0	20.0	31.6	20.0	67.9	12.7	14.0	17.2	10.1	26.8
	Tent	0.2	0.3	0.2	0.1	0.1	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1	0.0	0.1	0.1
	Tent	0.5	0.7	0.5	0.2	0.1	0.1	0.1	0.2	0.1	0.1	0.0	0.1	0.1	0.1	0.2	0.2
	EATA	0.5	1.1	1.7	0.7	0.9	1.8	0.8	0.5	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.6
	SAR	55.5	27.4	41.9	45.3	31.2	29.0	51.8	27.4	45.1	37.9	68.3	51.2	17.0	16.6	15.1	37.4
	MECTA	25.7	13.9	20.7	0.2	12.4	5.5	17.7	2.8	10.1	4.2	17.4	7.3	6.3	3.3	2.5	10.0
	Ours	51.7	25.2	36.8	12.6	35.5	28.5	55.9	13.3	42.7	30.6	72.7	22.4	9.6	27.4	14.9	32.0

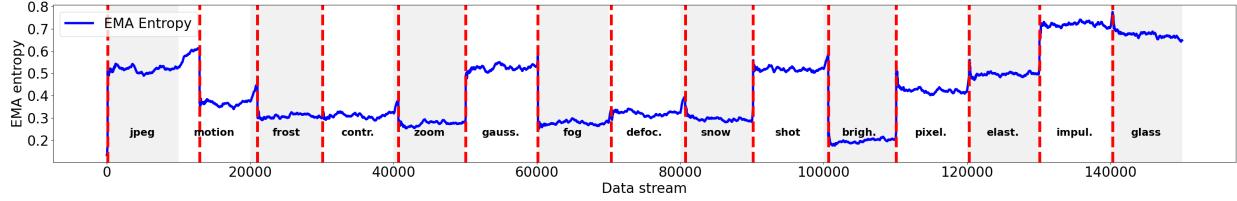


Figure 13. EMA entropy change along data stream on the random sequence. The red dotted lines are where domain shift is detected. Domains change after every 10,000 samples, as denoted by the changes in the background color.

to the minimal accuracy drop during this transition, as the model’s accuracy decreases only slightly from 75% to 66%. The use of EMA entropy as a detection signal proves to be an effective and lightweight approach, suitable for real-time TTA in resource-constrained environments.