Efficient Test-Time Model Adaptation without Forgetting

Shuaicheng Niu [* 1 2]   **Jiaxiang Wu** [* 3]   **Yifan Zhang** [* 4]   **Yaofo Chen** [1]
Shijian Zheng [1]   **Peilin Zhao** [3]   **Mingkui Tan** [1 5]

Abstract

Test-time adaptation (TTA) seeks to tackle potential distribution shifts between training and testing data by adapting a given model w.r.t. any testing sample. This task is particularly important for deep models when the test environment changes frequently. Although some recent attempts have been made to handle this task, we still face two practical challenges: 1) existing methods have to perform backward computation for each test sample, resulting in unbearable prediction cost to many applications; 2) while existing TTA solutions can significantly improve the test performance on out-of-distribution data, they often suffer from severe performance degradation on in-distribution data after TTA (known as catastrophic forgetting). In this paper, we point out that not all the test samples contribute equally to model adaptation, and high-entropy ones may lead to noisy gradients that could disrupt the model. Motivated by this, we propose an active sample selection criterion to identify reliable and non-redundant samples, on which the model is updated to minimize the entropy loss for test-time adaptation. Furthermore, to alleviate the forgetting issue, we introduce a Fisher regularizer to constrain important model parameters from drastic changes, where the Fisher importance is estimated from test samples with generated pseudo labels. Extensive experiments on CIFAR-10-C, ImageNet-C, and ImageNet-R verify the effectiveness of our proposed method. Code is available at https://github.com/mr-eggplant/EATA.

---
*Equal contribution. This work is done when Shuaicheng Niu works as an intern in Tencent AI Lab. [1]School of Software Engineering, South China University of Technology, China [2]Pazhou Laboratory, China [3]Tencent AI Lab, China [4]National University of Singapore, Singapore [5]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, China. Correspondence to: Mingkui Tan <mingkuitan@scut.edu.cn>.

1. Introduction

Deep neural networks (DNNs) have achieved excellent performance in many challenging tasks, including image classification (He et al., 2016), video recognition (Wang et al., 2018; Zeng et al., 2020; 2021; Chen et al., 2021), and many other areas (Choi et al., 2018; Fan et al., 2020; Xu et al., 2021). One prerequisite behind the success of DNNs is that the test samples are drawn from the same distribution as the training data, which, however, is often violated in many real-world applications. In practice, test samples may encounter natural variations or corruptions (also called *distribution shift*), such as changes in lighting resulting from weather change and unexpected noises resulting from sensor degradation (Hendrycks & Dietterich, 2019; Koh et al., 2021). Unfortunately, models are often very sensitive to such distribution shifts and suffer severe performance degradation.

Recently, several attempts (Sun et al., 2020; Wang et al., 2021; Liu et al., 2021; Zhang et al., 2021b;c; Wang et al., 2022) have been proposed to handle the distribution shifts by online adapting a model at test time (called *test-time adaptation*). Test-time training (TTT) (Sun et al., 2020) first proposes this pipeline. Given a test sample, TTT first fine-tunes the model via rotation classification (Gidaris et al., 2018) and then makes a prediction using the updated model. However, TTT still relies on additional training modifications (adding rotation head into the model), and thus the access to original training data is also compulsory. These requirements may be infeasible if, *e.g.*, the training data is unavailable due to privacy/storage concerns or the training involves unexpected heavy computation. To avoid these, Tent (Wang et al., 2021) and MEMO (Zhang et al., 2021b) propose methods for fully test-time adaptation, in which the adaptation only involves test samples and a trained model.

Although recent test-time adaptation methods are effective at handling test shifts, they still suffer the following limitations. First, since we adapt models at test time, the adaptation efficiency is quite important in many latency-sensitive scenarios. However, prior methods rely on performing backward computation for each test sample (even multiple backward passes for a single sample, such as TTT (Sun et al., 2020) and MEMO (Zhang et al., 2021b)). As performing back-propagation too much is time-consuming, these ap-

Table 1. Characteristics of problem settings that adapt a trained model to a potentially shifted test domain. 'Offline' adaptation assumes access to the entire source or target dataset, while 'Online' adaptation can predict a single or batch of incoming test samples immediately.

Setting	Source Data	Target Data	Training Loss	Testing Loss	Offline	Online	Source Acc.
Fine-tuning	✗	$\mathbf{x}^t, y^t$	$\mathcal{L}(\mathbf{x}^t, y^t)$	–	✓	✗	Not Considered
Continual learning	✗	$\mathbf{x}^t, y^t$	$\mathcal{L}(\mathbf{x}^t, y^t)$	–	✓	✗	Maintained
Unsupervised domain adaptation	$\mathbf{x}^s, y^s$	$\mathbf{x}^t$	$\mathcal{L}(\mathbf{x}^s, y^s) + \mathcal{L}(\mathbf{x}^s, \mathbf{x}^t)$	–	✓	✗	Maintained
Test-time training	$\mathbf{x}^s, y^s$	$\mathbf{x}^t$	$\mathcal{L}(\mathbf{x}^s, y^s) + \mathcal{L}(\mathbf{x}^s)$	$\mathcal{L}(\mathbf{x}^t)$	✗	✓	Not Considered
Fully test-time adaptation (FTTA)	✗	$\mathbf{x}^t$	✗	$\mathcal{L}(\mathbf{x}^t)$	✗	✓	Not Considered
EATA (ours)	✗	$\mathbf{x}^t$	✗	$\mathcal{L}(\mathbf{x}^t)$	✗	✓	Maintained

proaches may be infeasible when the latency is unacceptable. Second, these methods focus on boosting the performance of a trained model on out-of-distribution (OOD) test samples, ignoring that the model after test-time adaptation suffers a severe performance degradation (named *forgetting*) on in-distribution (ID) test samples (see Figure 3). An eligible test-time adaptation approach should perform well on both OOD and ID test samples simultaneously, since test samples actually often come with both ID and OOD domains.

To address above limitations, we propose an Efficient Anti-forgetting Test-time Adaptation method (namely EATA), which consists of a sample-efficient optimization strategy and a weight regularizer. Specifically, we devise a sample-adaptive entropy minimization loss, in which we exclude two types of samples out of optimization: i) samples with high entropy values, since the gradients provided by those samples are highly unreliable; and ii) samples are very similar. In this case, the total number of backward updates of a test data streaming is properly reduced (improving efficiency) and the model performance on OOD samples is also improved. On the other hand, we devise an anti-forgetting regularizer to enforce the important weights of the model do not change a lot during the adaptation. We calculate the weight importance based on Fisher information (Kirkpatrick et al., 2017) via a small set of test samples. With this regularization, the model can be continually adapted without performance degradation on ID test samples.

Contributions: 1) We propose an active sample identification scheme to filter out non-reliable and redundant test data from model adaptation; 2) We extend the label-dependent Fisher regularizer to test samples with pseudo label generation, which prevents drastic changes in important model weights; and 3) We demonstrate that the proposed EATA improves the efficiency of test-time adaptation and also alleviates the long-neglected catastrophic forgetting issue.

2. Related Work

We divide the discussion on related works based on the different adaptation settings summarized in Table 1.

Test-Time Adaptation (TTA) aims to improve model accuracy on OOD test data through model adaptation with test samples. Existing test-time training methods, *e.g.*, TTT (Sun et al., 2020), TTT++ (Liu et al., 2021) and MT3 (Bartler et al., 2022), jointly train a source model via both supervised and self-supervised objectives, and then adapt the model via self-supervised objective at test time. This pipeline, however, has assumptions on the manner of model training, which may not always be controllable in practice. To address this, fully test-time adaptation methods have been proposed to adapt a model with only test data, including batchnorm statistics adaptation (Nado et al., 2020; Schneider et al., 2020; Khurana et al., 2021), test-time entropy minimization (Wang et al., 2021; Fleuret et al., 2021), prediction consistency maximization over different augmentations (Zhang et al., 2021b), and classifier adjustment (Iwasawa & Matsuo, 2021). Our work follows the fully test-time adaptation setting and seeks to address two key limitations of prior works (i.e., efficiency hurdle and catastrophic forgetting) to make TTA more practical in real-world applications.

Continual Learning (CL) aims to help the model remember the essential concepts that have been learned previously, alleviating the catastrophic forgetting issue when learning on a new task (Kirkpatrick et al., 2017; Li & Hoiem, 2017; Rolnick et al., 2019; Farajtabar et al., 2020; Niu et al., 2021; Mittal et al., 2021). In our work, we share the same motivation as CL and point out that test-time adaptation also suffers catastrophic forgetting (*i.e.*, performance degradation on ID test samples), which makes test-time adaptation approaches are unstable to deploy. To conquer this, we propose a simple yet effective solution to maintain the model performance on ID test samples (by only using test data) and meanwhile improve the performance on OOD test samples.

Unsupervised Domain Adaptation (UDA). Conventional UDA tackles distribution shifts by jointly optimizing a source model on both labeled source data and unlabeled target data, such as devising a domain discriminator to learn domain-invariant features (Pei et al., 2018; Saito et al., 2018; Zhang et al., 2020b;a). To avoid access to source data, recent source-free UDA methods are proposed either by generative
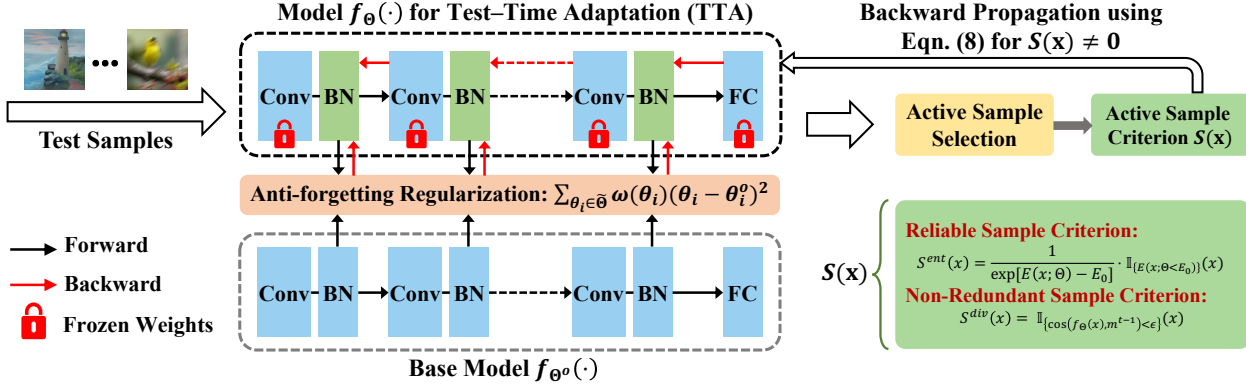
Figure 1. An illustration of the proposed EATA. Given a trained base model f_{Θ^o}, we perform test-time adaptation with a model f_Θ that initialized from Θ^o. During the adaptation process, we only update the parameters of batch normalization layers in f_Θ and froze the rest parameters. When a batch of test sample $\mathcal{X} = \{\mathbf{x}_b\}_{b=1}^B$ come, we calculate a sample-adaptive weight $S(\mathbf{x})$ for each test sample to identify whether the sample is active for adaptation or not. We only perform backward propagation with the samples whose $S(\mathbf{x}) \neq 0$. Moreover, we propose an anti-forgetting regularizer to prevent the model parameters Θ changing too much from Θ^o.

modeling (Li et al., 2020; Kundu et al., 2020; Qiu et al., 2021) or information maximization (Liang et al., 2020). Nevertheless, such methods optimize offline via multiple epochs and losses. In contrast, our method adapts in an online manner and selectively performs once backward propagation for one given target sample, which is more efficient during inference.

3. Problem Formulation

Without loss of generality, let $P(\mathbf{x})$ be the distribution of training data $\{\mathbf{x}_i\}_{i=1}^N$ (namely $\mathbf{x}_i \sim P(\mathbf{x})$) and $f_{\Theta^o}(\mathbf{x})$ be a **base model** trained on labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where Θ^o denotes the model parameters. Due to the training process, the model $f_{\Theta^o}(\mathbf{x})$ tends to fit (or overfit) the training data. During the inference state, the model shall perform well for the in-distribution test data, namely $\mathbf{x} \sim P(\mathbf{x})$. However, in practice, due to possible distribution shifts between training and test data, we may encounter many out-of-distribution test samples, namely $\mathbf{x} \sim Q(\mathbf{x})$, where $Q(\mathbf{x}) \neq P(\mathbf{x})$. In this case, the prediction would be very unreliable and the performance is also very poor.

Test-time adaptation (TTA) (Wang et al., 2021; Zhang et al., 2021b) aims at boosting the out-of-distribution prediction performance by doing model adaptation on test data only. Specifically, given a set of test samples $\{\mathbf{x}_j\}_{j=1}^M$, where $\mathbf{x}_j \sim Q(\mathbf{x})$ and $Q(\mathbf{x}) \neq P(\mathbf{x})$, one needs to adapt $f_\Theta(\mathbf{x})$ to improve the prediction performance on test data in any cases. To achieve this, existing methods often seek to update the model by minimizing some unsupervised objective defined on test samples:

$$\min_{\tilde{\Theta}} \mathcal{L}(\mathbf{x}; \Theta), \ \mathbf{x} \sim Q(\mathbf{x}), \tag{1}$$

where $\tilde{\Theta} \subseteq \Theta$ denotes the free model parameters that should

be updated. In general, the test-time learning objective $\mathcal{L}(\cdot)$ can be formulated as an entropy minimization problem (Wang et al., 2021) or prediction consistency maximization over data augmentations (Zhang et al., 2021b).

For existing TTA methods like TTT (Sun et al., 2020) and MEMO (Zhang et al., 2021b)), during the test-time adaptation, we shall need to compute one round or even multiple round of backward computation for each sample, which is very time-consuming and also not favorable for latency-sensitive applications. Moreover, most methods assume that all the test samples are drawn from out-of-distribution (OOD). However, in practice, the test samples may come from both in-distribution (ID) and OOD. Simply optimizing the model on OOD test samples may lead to severe performance degradation on ID test samples. We empirically validate the existence of such issue in Figure 3, where the updated model has a consistently lower accuracy on ID test samples than the original model.

4. Proposed Methods

In this paper, we propose an Efficient Anti-forgetting Test-time Adaptation (EATA) method, which aims to improve the efficiency of test-time adaptation (TTA) and tackle the catastrophic forgetting issue brought by existing TTA strategies simultaneously. As shown in Figure 1, EATA consists of two strategies. **1)** *Sample-efficient entropy minimization* (c.f. Section 4.1) aims to conduct efficient adaptation relying on an active sample selection strategy. Here, the sample selection process is to choose only active samples for backward propagation and therefore improve the overall TTA efficiency (*i.e.*, less gradient backward propagation). To this end, we devise an active sample selection score, denoted by $S(\mathbf{x})$, to detect those reliable and non-redundant test

Algorithm 1 The pipeline of proposed EATA.

Input: Test samples $\mathcal{D}_{test}=\{\mathbf{x}_j\}_{j=1}^M$, the trained model $f_\Theta(\cdot)$, ID samples $\mathcal{D}_F=\{\mathbf{x}_q\}_{q=1}^Q$, batch size B.

1: **for** a batch $\mathcal{X}=\{\mathbf{x}_b\}_{b=1}^B$ in $\mathcal{D}_{test}$ **do**
2:     Calculate predictions $\hat{y}$ for all $\mathbf{x} \in \mathcal{X}$ via $f_\Theta(\cdot)$.
3:     Calculate sample selection score $S(\mathbf{x})$ via Eqn. (6).
4:     Update model ($\tilde{\Theta} \subseteq \Theta$) with Eqn. (2) or Eqn. (8).
5: **end for**

Output: The predictions $\{\hat{y}\}_{j=1}^M$ for all $\mathbf{x} \in \mathcal{D}_{test}$.

Figure 2. Effect of different test samples in test-time entropy minimization (Wang et al., 2021). We adapt a model on partial samples (top $p\%$ samples with the highest or lowest entropy values), and then evaluate the adapted model on all test samples. Results are obtained on ImageNet-C (Gaussian noise, level 3) and ResNet-50 (base accuracy is 27.6%). Introducing more samples with high entropy values into adaptation will hurt the adaptation performance.

samples from the test set for TTA. **2)** *Anti-forgetting weight regularization* (c.f. Section 4.2) seeks to alleviate knowledge forgetting by enforcing that the parameters, important for the ID domain, do not change too much in test-time adaptation. In this way, the catastrophic forgetting issue can be significantly alleviated. The pseudo-code of EATA is summarized in Algorithm 1.

4.1. Sample Efficient Entropy Minimization

For efficient test-time adaptation, we propose an active sample identification strategy to select samples for backward propagation. Specifically, we design an active sample selection score for each sample, denoted by $S(\mathbf{x})$, based on two criteria: 1) samples should be **reliable** for test-time adaptation, and 2) the samples involved in optimization should be **non-redundant**. By setting $S(\mathbf{x})=0$ for non-active samples, namely the unreliable and redundant samples, we can reduce unnecessary backward computation during test-time adaptation, thereby improving the prediction efficiency.

Relying on the sample score $S(\mathbf{x})$, following (Wang et al., 2021; Zhang et al., 2021b), we use entropy loss for model adaptations. Then, the sample-efficient entropy minimization is to minimize the following objective:

$$\min_{\tilde{\Theta}} S(\mathbf{x})E(\mathbf{x};\Theta) = -S(\mathbf{x})\sum_{y\in\mathcal{C}} f_\Theta(y|\mathbf{x})\log f_\Theta(y|\mathbf{x}), \quad (2)$$

where $\mathcal{C}$ is the model output space. Here, the entropy loss $E(\cdot)$ is computed over a batch of samples each time (similar to Tent (Wang et al., 2021)) to avoid a trivial solution, *i.e.*, assigning all probability to the most probable class. For efficient adaptation, we update $\tilde{\Theta} \subseteq \Theta$ with the affine parameters of all batch normalization layers.

Reliable Sample Identification. Our intuition is that different test samples produce various effects in adaptation. To verify this, we conduct a preliminary study, where we select different proportions of samples (the samples are pre-sorted according to their entropy values $E(\mathbf{x};\Theta)$) for adaptation, and the resulting model is evaluated on all test samples. From Figure 2, we find that: 1) adaptation on low-entropy samples makes more contribution than high-entropy ones, and 2) adaptation on test samples with very
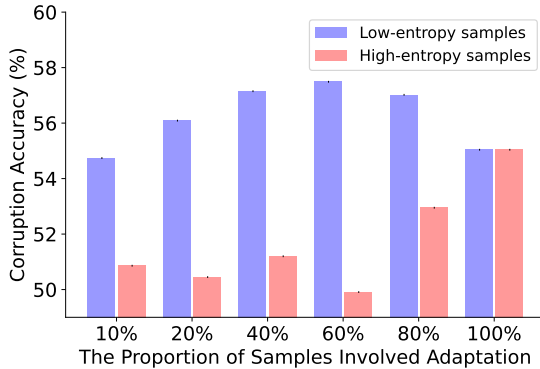
high entropy may hurt performance. The possible reason is that predictions of high-entropy samples are uncertain, so their gradients produced by entropy loss may be biased and unreliable. Following this, we name these low-entropy samples as reliable samples.

Based on the above observation, we propose an entropy-based weighting scheme to identify reliable samples and emphasize their contributions during adaptation. Formally, the entropy-based weight is given by:

$$S^{ent}(\mathbf{x}) = \frac{1}{\exp\left[E(\mathbf{x};\Theta) - E_0\right]} \cdot \mathbb{I}_{\{E(\mathbf{x};\Theta)<E_0\}}(\mathbf{x}), \quad (3)$$

where $\mathbb{I}_{\{\cdot\}}(\cdot)$ is an indicator function, $E(\mathbf{x};\Theta)$ is the entropy of sample $\mathbf{x}$, and E_0 is a pre-defined threshold. The above weighting function excludes high-entropy samples from adaptation and assigns higher weights to test samples with lower prediction uncertainties, allowing them to contribute more to model updates. Note that evaluating $S^{ent}(\mathbf{x})$ does not involve any gradient back-propagation.

Non-redundant Sample Identification. Although Eqn. (3) helps to exclude partial unreliable samples, the remaining test samples may still have redundancy. For example, given two test samples that are mutually similar and both have a lower prediction entropy than E_0, we still need to perform gradient back-propagation for each of them according to Eqn. (3). However, this is unnecessary as these two samples produce similar gradients for model adaptation.

To further improve efficiency, we propose to exploit the samples that can produce different gradients for model adaptation. Recall that the entropy loss only relies on final model outputs (*i.e.*, classification logits), we further filter samples by ensuring the remaining samples have diverse model out-

puts. To this end, one straightforward method is to save the model outputs of all previously seen samples, and then compute the similarity between the outputs of incoming test samples and all saved model outputs for filtering. However, this method is computational expensive at test time and memory-consuming with the increase of test samples.

To address this, we exploit an exponential moving average technique to track the average model outputs of all seen test samples used for model adaptation. To be specific, given a set of model outputs of test samples, the moving average vector is updated recursively:

$$\mathbf{m}^t = \begin{cases} \bar{\mathbf{y}}^1, & \text{if } t = 1 \\ \alpha \bar{\mathbf{y}}^t + (1-\alpha)\mathbf{m}^{t-1}, & \text{if } t > 1 \end{cases}, \quad (4)$$

where $\bar{\mathbf{y}}^t = \frac{1}{n}\sum_{k=1}^{n}\hat{\mathbf{y}}_k^t$ is the average model prediction of a mini-batch of n test samples at the iteration t, and $\alpha \in [0,1]$. Following that, given a new test sample $\mathbf{x}$ received at iteration $t > 1$, we compute the cosine similarity between its prediction $f_\Theta(\mathbf{x})$ and the moving average $\mathbf{m}^{t-1}$ (i.e., $cos(f_\Theta(\mathbf{x}), \mathbf{m}^{t-1})$), which is then used to determine the diversity-based weight:

$$S^{div}(\mathbf{x}) = \mathbb{I}_{\{cos(f_\Theta(\mathbf{x}), \mathbf{m}^{t-1}) < \epsilon\}}(\mathbf{x}), \quad (5)$$

where ϵ is a pre-defined threshold for cosine similarities. The overall sample-adaptive weight is then given by:

$$S(\mathbf{x}) = S^{ent}(\mathbf{x}) \cdot S^{div}(\mathbf{x}), \quad (6)$$

which combines both entropy-based (as in Eqn. 3) and diversity-based terms (as in Eqn. 5). Since we only perform gradient back-propagation for test samples with $S(\mathbf{x}) > 0$, the algorithm efficiency is further improved.

Remark. Given M test samples $\mathcal{D}_{test} = \{\mathbf{x}_j\}_{j=1}^{M}$, the total number of reduced backward computation is given by $\mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{test}}[\mathbb{I}_{\{S(\mathbf{x})=0\}}(\mathbf{x})]$, which is jointly determined the test data distribution $\mathcal{D}_{test}$, entropy threshold E_0, and cosine similarity threshold ϵ.

4.2. Anti-Forgetting with Fisher Regularization

In this section, we propose a new weighted Fisher regularizer (called anti-forgetting regularizer) to alleviate the catastrophic forgetting issue caused by test-time adaptation, i.e., the performance of a test-time adapted model may significantly degrade on in-distribution (ID) test samples. We achieve this through weight regularization, which only affects the loss function and does not incur any additional computational overhead for model adaptation. To be specific, we apply an importance-aware regularizer $\mathcal{R}$ to prevent model parameters, important for the in-distribution domain, from changing too much during the test-time adaptation process (Kirkpatrick et al., 2017):

$$\mathcal{R}(\tilde{\Theta}, \tilde{\Theta}^o) = \sum_{\theta_i \in \tilde{\Theta}} \omega(\theta_i)(\theta_i - \theta_i^o)^2, \quad (7)$$

where $\tilde{\Theta}$ are parameters used for model update and $\tilde{\Theta}^o$ are the corresponding parameters of the original model. $\omega(\theta_i)$ denotes the importance of θ_i and we measure it via the diagonal Fisher information matrix as in elastic weight consolidation (Kirkpatrick et al., 2017). Here, the calculation of Fisher information $\omega(\theta_i)$ is non-trivial since we are inaccessible to any labeled training data. For the convenience of presentation, we leave the details of calculating $\omega(\theta_i)$ in the next subsection.

After introducing the anti-forgetting regularizer, the final optimization formula for our method can be formulated as:

$$\min_{\tilde{\Theta}} S(\mathbf{x})E(\mathbf{x};\Theta) + \beta \mathcal{R}(\tilde{\Theta}, \tilde{\Theta}^o), \quad (8)$$

where β is a trade-off parameter, $S(\mathbf{x})$ and $E(\mathbf{x};\Theta)$ are defined in Eqn. (2).

Measurement of Weight Importance $\omega(\theta_i)$. The calculation of Fisher information typically involves a set of labeled ID training samples. However, in our problem setting, we are inaccessible to training data and the test samples are only unlabeled, which makes it non-trivial to measure the weight importance. To conquer this, we first collect a small set of unlabeled ID test samples $\{\mathbf{x}_q\}_{q=1}^{Q}$, and then use the original trained model $f_\Theta(\cdot)$ to predict all these samples for obtaining the corresponding hard pseudo-label $\hat{y}_q$. Following that, we construct a pseudo-labeled ID test set $\mathcal{D}_F = \{\mathbf{x}_q, \hat{y}_q\}_{q=1}^{Q}$, based on which we calculate the fisher importance of model weights by:

$$\omega(\theta_i) = \frac{1}{Q} \sum_{\mathbf{x}_q \in \mathcal{D}_F} \left(\frac{\partial}{\partial \theta_i^o} \mathcal{L}_{CE}(f_{\Theta^o}(\mathbf{x}_q), \hat{y}_q)\right)^2, \quad (9)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss. Here, we only need to calculate $\omega(\theta_i)$ once before performing test-time adaptation. Once calculated, we keep $\omega(\theta_i)$ fixed and apply it to any types of distribution shifts. Moreover, the unlabeled ID test samples are collected based on out-of-distribution detection techniques (Liu et al., 2020; Berger et al., 2021), which are easy to implement. Note that there is no need to collect too many ID test samples for calculating $\omega(\theta_i)$, e.g., 500 samples are enough for ImageNet-C dataset. More empirical studies regarding this can be found in Figure 4(b).

5. Experiments

We organize the experiments to answer following questions: 1) How does EATA compare with prior methods regarding efficiency and accuracy? 2) Can EATA alleviate the forgetting that occurred after test-time adaptation? 3) How do different components affect the performance of EATA?

Datasets and Models. We conduct experiments on three benchmarks datasets for OOD generalization, i.e., CIFAR-10-C, ImageNet-C (Hendrycks & Dietterich, 2019) (contains corrupted images in 15 types of 4 main categories and

Table 2. Comparison with state-of-the-art methods on ImageNet-C with the highest severity level 5 regarding **Corruption Error (%, ↓)**. "GN" and "BN" denote group and batch normalization, respectively. "JT" denotes the model is jointly trained via supervised cross-entropy and rotation prediction losses. The **bold** number indicates the best result and the <u>underlined</u> number indicates the second best result.

	Noise			Blur				Weather				Digital				Average	
Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	#Forwards	#Backwards
R-50 (GN)+JT	94.9	95.1	94.2	88.9	91.7	86.7	81.6	82.5	81.8	80.6	49.2	87.4	76.9	79.2	68.5	50,000	0
• TTT	69.0	66.4	66.6	71.9	92.2	66.8	63.2	59.1	81.0	49.0	38.2	61.1	50.6	48.3	52.0	50,000×21	50,000×20
R-50 (BN)	97.8	97.1	98.2	82.1	90.2	85.2	77.5	83.1	76.7	75.6	41.1	94.6	83.1	79.4	68.4	50,000	0
• TTA	95.9	95.1	95.5	87.5	91.8	87.1	74.2	86.0	80.9	78.7	47.0	87.6	85.4	75.4	66.4	50,000×64	0
• BN adaptation	84.5	83.9	83.7	80.0	80.0	71.5	60.0	65.2	65.0	51.5	34.1	75.9	54.2	49.3	58.9	50,000	0
• MEMO	92.5	91.3	91.0	80.3	87.0	79.3	72.4	74.7	71.2	67.9	39.0	89.0	76.2	67.0	62.5	50,000×65	50,000×64
• Tent	71.6	69.8	69.9	71.8	72.7	58.6	50.5	52.9	58.7	42.5	32.6	74.9	45.2	41.5	47.7	50,000	50,000
• Tent (episodic)	85.4	84.8	84.9	85.5	85.4	74.6	62.2	66.4	67.8	53.2	35.7	83.9	57.1	52.4	61.5	50,000×2	50,000
• ETA (ours)	**64.9**	<u>62.1</u>	<u>63.4</u>	66.1	67.1	**52.2**	47.4	**48.1**	54.2	39.9	32.1	**55.0**	42.1	39.1	<u>45.1</u>	50,000	26,031
• EATA (ours)	<u>65.0</u>	63.1	64.3	66.3	<u>66.6</u>	52.9	<u>47.2</u>	<u>48.6</u>	54.3	40.1	**32.0**	<u>55.7</u>	<u>42.4</u>	<u>39.3</u>	**45.0**	50,000	25,150
• EATA (lifelong)	<u>65.0</u>	**61.9**	**63.2**	<u>66.2</u>	**65.8**	<u>52.7</u>	**46.8**	48.9	54.4	40.3	**32.0**	55.8	42.8	39.6	45.3	50,000	28,243

each type has 5 severity levels) and ImageNet-R (Hendrycks et al., 2021). We use ResNet-26 (R-26)/ResNet-50 (R-50) (He et al., 2016) for CIFAR-10/ImageNet experiments. The models are trained on CIFAR-10 or ImageNet training set and then tested on clean or the above OOD test sets.

Compared Methods. We compare with following state-of-the-art methods. Test-time Training (TTT) (Sun et al., 2020) adapts a model via rotation prediction at test time, but requires the model also being trained by rotation prediction. Test-time Augmentation (TTA) (Ashukha et al., 2020) predicts a sample via the average outputs of its different augmentations. BN adaptation (Schneider et al., 2020) updates batch normalization statistics on test samples. Tent (Wang et al., 2021) minimizes the entropy of test samples during testing. MEMO (Zhang et al., 2021b) maximizes the prediction consistency of different augmented copies regarding a given test sample. We denote EATA without weight regularization in Eqn. (7) as **e**fficient **t**est-time **a**daptation (ETA).

For TTT, Tent, our ETA and EATA, the model is online adapted through the entire evaluation of one given test dataset (*e.g.*, gaussian noise level 5 of ImageNet-C). Once the adaptation on this dataset is finished, the model parameters will be reset. For TTT (episodic) and Tent (episodic), the model parameters will be reset immediately after each optimization step of a test sample or batch. For EATA (lifelong) and Tent (lifelong), the model is online adapted and the parameters will never be reset (as shown in Figure 3 (**Right**)), which is more challenging but practical.

Evaluation Metrics. 1) Clean accuracy/error (%) on original in-distribution (ID) test samples, *e.g.*, the original test images of ImageNet. Note that we measure the clean accuracy of all methods via **(re)adaptation**; 2) Out-of-distribution (OOD) accuracy/error (%) on OOD test samples, *e.g.*, the corruption accuracy on ImageNet-C; 3) The number of forward and backward passes during the entire test-time adaptation process. Note that the fewer #forwards and #backwards indicate the less computation, leading to higher efficiency.

Table 3. Comparison on ImageNet-R. The base model is ResNet-50 (using batchnorm) trained on original ImageNet training set.

Model	Error (%)	#Forwards	#Backwards
Base Model	63.8	30,000	0
• TTA (Ashukha et al., 2020)	$61.3_{(-2.5)}$	30,000×64	0
• BN (Schneider et al., 2020)	$59.7_{(-4.1)}$	30,000	0
• MEMO (Zhang et al., 2021b)	$58.8_{(-5.0)}$	30,000×65	30,000×64
• Tent (Wang et al., 2021)	$57.7_{(-6.1)}$	30,000	30,000
• Tent (episodic)	$61.0_{(-2.9)}$	30,000×2	30,000
• ETA (ours)	$\mathbf{54.5}_{(-9.3)}$	30,000	14,847
• EATA (ours)	$54.8_{(-9.0)}$	30,000	14,800

Implementation Details. For test-time adaptation, we use SGD as the update rule, with a momentum of 0.9, batch size of 64, and learning rate of 0.005/0.00025 for CIFAR-10/ImageNet (following Tent and MEMO). The entropy constant E_0 in Eqn. (3) is set to $0.4 \times \ln C$, where C is number of task classes. The ϵ in Eqn. (6) is set to 0.4/0.05 for CIFAR-10/ImageNet. The trade-off parameter β in Eqn. (8) is set to 1/2,000 for CIFAR-10/ImageNet to make two losses have the similar magnitude. We use 2,000 samples to calculate $\omega(\theta_i)$ in Eqn. (9) which takes 2,000 additional forward-and-backward passes. The moving average factor α in Eqn. (4) is set to 0.1. More details are put in Supplementary.

5.1. Comparisons of OOD Performance and Efficiency

Results on ImageNet-C. We report the comparisons on ImageNet-C with the highest severity level 5 in Table 2 and put more results of other severity levels 1-4 into Supplementary due to the page limitation. From the results, our ETA and EATA consistently outperform the considered methods in all 15 corruption types regarding the classification error, suggesting our effectiveness. With our sample-adaptive entropy loss, ETA achieves a large performance gain over Tent (*e.g.*, $71.6\% \rightarrow 65.0\%$ on Gaussian noise), verifying that removing samples with unreliable gradients and tackling samples differently benefits the test-time adaptation. More critically, ETA outperforms TTT consistently (while Tent
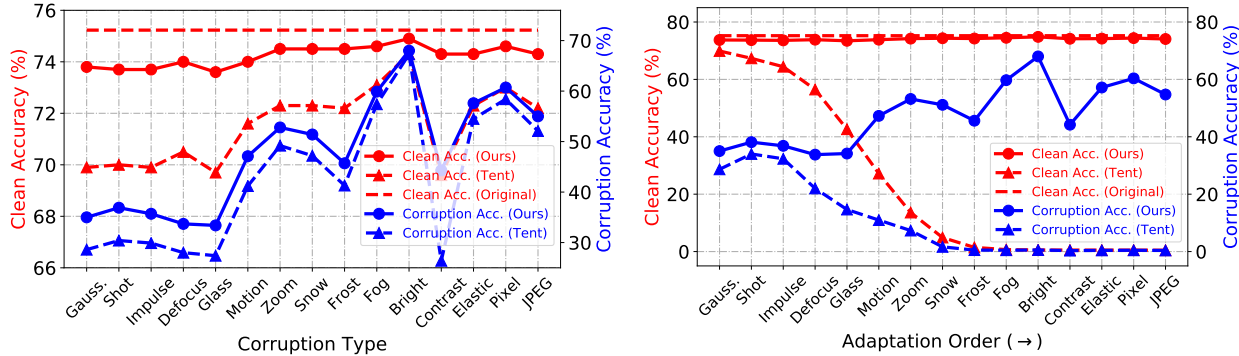
Figure 3. Comparison of prevent forgetting on ImageNet-C (severity level 5) with ResNet-50. We record the OOD corruption accuracy on each corrupted test set and the ID clean accuracy (after OOD adaptation). In **Left**, the model parameters of both Tent and our EATA are reset before adapting to a new corruption type. In **Right**, the model performs lifelong adaptation and the parameters will never be reset, namely Tent (lifelong) and our EATA (lifelong). EATA achieves higher OOD accuracy and meanwhile maintains the ID clean accuracy.

Table 4. Comparison on CIFAR-10-C. Each result is averaged over 15 different corruption types and 5 severity levels (totally 75).

Model	Average Error (%)	Average #Forwards	Average #Backwards
ResNet-26 (GroupNorm)	22.5	10,000	0
• TTA (Ashukha et al., 2020)	$19.9_{(-2.6)}$	10,000×32	0
• MEMO (Zhang et al., 2021b)	$19.6_{(-2.9)}$	10,000×33	10,000×32
ResNet-26 (GroupNorm)+JT	22.8	10,000	0
• TTT (Sun et al., 2020)	$\mathbf{15.6}_{(-7.2)}$	10,000×33	10,000×32
• TTT (episodic)	$21.5_{(-1.3)}$	10,000×33	10,000×32
ResNet-26 (BatchNorm)	28.4	10,000	0
• Tent (Wang et al., 2021)	$20.2_{(-8.2)}$	10,000	10,000
• ETA (ours)	$19.4_{(-\mathbf{9.0})}$	10,000	8,192
• EATA (ours)	$19.7_{(-8.7)}$	10,000	8,153

fails to achieve this), demonstrating the potential of fully test-time adaptation methods, *i.e.*, boosting OOD generalization without altering the training process. Compared with ETA, EATA and EATA (lifelong) achieve comparable OOD performance (but prevent the forgetting on ID samples, see Figure 3), showing that our anti-forgetting regularization does not limit the learning ability during adaptation.

As for efficiency, the required average backward number of our ETA is 26,031, which is much fewer than those methods that need multiple data augmentations (*i.e.*, TTT and MEMO are 50,000×20 and 64). Compared with Tent, ETA reduces the average #backward from 50,000 to 26,031, by excluding samples with high prediction entropy and samples that are similar out of test-time optimization. In this sense, our method only needs to adapt for partial samples, resulting in higher efficiency. Note that although optimization-free methods (such as BN adaptation) do not need backward updates, their OOD performances are limited.
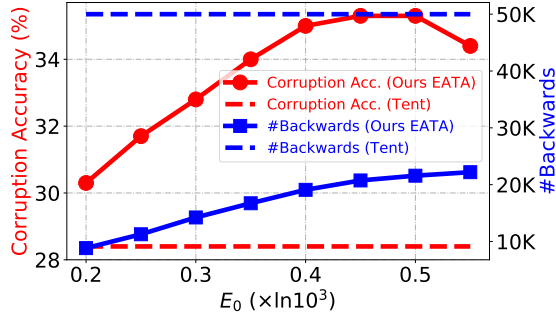
Results on ImageNet-R and CIFAR-10-C. From Table 3, our ETA yields 54.5% classification error on ImageNet-R

(-3.2% over the best counterpart method Tent) and only needs 14,847 backward propagation (much fewer than other learning-based test-time adaptation methods, *e.g.*, MEMO and Tent). The results on CIFAR-10-C are shown in Table 4. Under the same base model (ResNet-26 with batch normalization), ETA achieves lower average error than Tent (19.4% *vs.* 20.2%) with less requirements of back-propagation (8,192 *vs.* 10,000). Moreover, the performance gain over the base model of ETA is larger than that of TTT, *i.e.*, -9.0% *vs.* -7.2% average error. These results are consistent with the ones on ImageNet-C that ETA achieves higher performance and improves the efficiency, further demonstrating the effectiveness and superiority of our method.
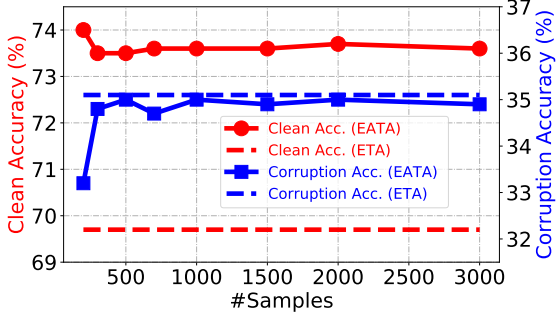
5.2. Demonstration of Preventing Forgetting

In this section, we investigate the ability of our EATA in preventing ID forgetting during test-time adaptation. The experiments are conducted on ImageNet-C with ResNet-50. We measure the anti-forgetting ability by comparing the model's clean accuracy (*i.e.*, on original validation data of ImageNet) before and after adaptation. To this end, we first perform test-time adaptation on a given OOD test set, and then evaluate the clean accuracy of the updated model. Here, we consider two adaptation scenarios: i) the model parameters will be reset before adapting to a new corrupted test set; 2) the model parameters will never be reset (namely lifelong adaptation), which is more challenging but practical. We report the results of severity level 5 in Figure 3 and put results of severity levels 1-4 into Supplementary.

From Figure 3, our EATA consistently outperforms Tent regarding the OOD corruption accuracy and meanwhile maintains the clean accuracy on ID data (in both two adaptation scenarios), demonstrating our effectiveness. It is worth noting that the performance degradation in lifelong adaptation scenario is much more severe (see Figure 3 **Right**). More

(a) Effect of different entropy margins E_0 in Eqn. (3)

(b) Effect of #samples for calculating Fisher in Eqn. (9).

Figure 4. Ablation experiments on ImageNet-C (Gaussian noise, severity level=5) with ResNet-50.

Table 5. Effectiveness of components in sample-adaptive weight $S(\mathbf{x})$ in EATA on ImageNet-C (Gaussian noise) with ResNet-50.

Method	Level 3		Level 5	
	Error (%)	#Backwards	Error (%)	#Backwards
Baseline ($S(\mathbf{x})$=1)	45.3	50,000	71.6	50,000
$+S^{ent}(\mathbf{x})$ (Eqn. 3)	43.0	37,943	65.7	26,694
$+S(\mathbf{x})$ (Eqn. 6)	**42.6**	**29,051**	**65.0**	**19,121**

vs. 65.7%), demonstrating the effectiveness of our proposed sample-efficient optimization strategy.

Entropy Constant E_0 in Eqn. (3). We evaluate our EATA with different E_0, selected from {0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55}$\times \ln 10^3$, where 10^3 is the class number of ImageNet. From Figure 4(a), our EATA achieves excellent performance when E_0 belongs to $[0.4, 0.5]$. Either a smaller or larger E_0 would hamper the performance. The reasons are mainly as follows. When E_0 is small, EATA removes too many samples during adaptation and thus is unable to learn enough adaptation knowledge from the remaining samples. When E_0 is too large, some high-entropy samples would take part in the adaptation but contribute unreliable and harmful gradients, resulting in performance degradation. As for larger E_0 leads to more backward passes, we set E_0 to $0.4\times \ln 10^3$ for the efficiency-performance trade-off and fix the proportion of 0.4 for all other experiments.

Number of Samples for Calculating Fisher in Eqn. (9). As described in Section 4.2, the calculation of Fisher information involves a small set of unlabeled ID samples, which can be collected via existing OOD detection techniques (Berger et al., 2021). Here, we investigate the effect of #samples Q, selected from {200, 300, 500, 700, 1000, 1500, 2000, 3000}. From Figure 4(b), our EATA achieves stable performance with $Q \geq 300$, *i.e.*, compared with ETA (without regularization), the OOD performance is comparable and the clean accuracy is much higher. These results show that our EATA does not need to collect too many ID samples, which are easy to obtain in practice.

5.4. More Discussions

Results on Single Sample Adaptation (Batch Size $B = 1$). In our proposed methods, we conduct test-time learning over a batch of test samples each time. When batch size B equals to 1, directly applying Tent and our ETA may fail (see Table 6), but it is not a very significant issue in practice. Actually, one can address this by either maintaining a sliding window $\{\mathbf{x}_i\}_{i=t-L}^{t}$ including L previous samples and the current test sample $\mathbf{x}_t$, or data augmentation techniques adopted in existing single sample TTA methods ((Zhang et al., 2021b; Khurana et al., 2021)). The first way is simple to implement and incurs only minor extra costs. From Table 6, our ETA with sliding window (ETA-wnd) works well and consistently outperforms the Tent counterpart.

critically, in lifelong adaptation, both the clean and corruption accuracy of Tent decreases rapidly (until degrades to 0%) after adaptation of the first three corruption types, showing that Tent in lifelong adaptation is not stable enough. In contrast, during the whole lifelong adaptation process, our EATA achieves good corruption accuracy and the clean accuracy is also very close to the clean accuracy of the model without any OOD adaptation (*i.e.*, original clean accuracy, tested using Tent). These results demonstrate the superiority of EATA in terms of overcoming the forgetting on ID data.

5.3. Ablation Studies

Effect of Components in $S(\mathbf{x})$ (Eqn. 6). Our EATA accelerates test-time adaptation by excluding two types of samples out of optimization: 1) samples with high prediction entropy values (Eqn. 3) and 2) samples that are similar (Eqn. 6). We ablate both of them in Table 5. Compared with the baseline $S(\mathbf{x})$=1 (the same as Tent), introducing $S^{ent}(\mathbf{x})$ in Eqn. (3) achieves lower error and fewer backwards (*e.g.*, 65.7% (26,694) *vs.* 71.6% (50,000) on level 5). This verifies our motivation in Figure 2 that some high-entropy samples may hurt the performance since their gradients are unreliable. When further removing some redundant samples that are similar (Eqn. 6), our EATA further reduces the number of back-propagation (*e.g.*, 26,694→19,121 on level 5) and achieves comparable OOD error (*e.g.*, 65.0%

Table 6. Effectiveness of ETA under sliding window strategy (with different window length L) for single sample adaptation. We report corruption accuracy (%) on ImageNet-C (Gaussian noise, level 5).

Base ResNet-50	Tent	ETA	Tent-wnd (L=32)	ETA-wnd (L=32)	Tent-wnd (L=64)	ETA-wnd (L=64)
2.2	0.1	0.1	28.1	30.8	29.5	**32.4**

Table 7. Comparison with Tent (Wang et al., 2021) w.r.t. corruption accuracy (%) with mixture of 15 corruption types on ImageNet-C.

Severity	Base (ResNet-50)	Tent	EATA (ours)
Level=3	39.8	$41.5_{(+1.7)}$	$\textbf{52.6}_{(+12.8)}$
Level=5	18.0	$2.3_{(-15.7)}$	$\textbf{26.6}_{(+8.6)}$

Table 8. Effectiveness of EATA with different backbone models. We report the corruption accuracy (%) on ImageNet-C (Gaussian noise, severity level 5).

Base	ResNet-50 (2.2)	ResNet-101 (3.5)	ResNet-152 (3.6)
Tent	$28.4_{(+26.2)}$	$32.5_{(+29.0)}$	$33.7_{(+30.1)}$
EATA	$\textbf{35.0}_{(+32.8)}$	$\textbf{38.8}_{(+35.3)}$	$\textbf{40.2}_{(+36.6)}$

Wall-clock Time Speed-up of EATA. In the current PyTorch version, gradient computation is conducted on the full mini-batch, even if instance-wise masks are applied. We achieve wall-time speed-up with one forward-only pass with B samples, and one forward-and-backward pass with B' samples selected by ETA/EATA. For ResNet-50 on ImageNet-C (Gaussian noise, level=5, 50,000 images) with one Tesla V100 GPU, the actual run time is 113s for Tent (28.6% accuracy) and 102s for EATA (35.1% accuracy). Actually, this is a more engineering problem, and an ideal implementation (forward with B samples and backward with B' samples) should further speed up the computation.

Additional Memory by Fisher Regularizer. Since we only regularize the affine parameters of BN layers, EATA needs very little extra memory. For ResNet-50 on ImageNet-C, the extra GPU memory at run time is only 9.8 MB, which is much less than that of Tent with batch size 64 (5,675 MB).

EATA under Mixed-and-Shifted Distributions. We evaluate Tent and our EATA on mixed ImageNet-C (level=3 or 5) that consists of 15 different corruption types/distribution shifts (totally 750k images). Results in Table 7 show the stability of EATA on large-scale and complex TTA scenarios.

EATA with Large Network Models. In Table 8, we apply EATA to pre-trained models with different computational complexities. From the results, even for large models (ResNet-101 and ResNet-152) that tend to show overconfident, our EATA still works well.

Table 9. Comparison with Tent (Wang et al., 2021) w.r.t. corruption accuracy (%) with fewer (N) test samples on ImageNet-C (Gaussian noise, severity level 5).

Method	N=256	N=512	N=1,024	N=2,048	N=4,096	N=10,000
Tent	13.7	18.2	14.8	15.2	17.5	20.9
EATA	**14.8**	**20.1**	**16.3**	**18.7**	**23.5**	**27.6**

EATA with Different Number of Test Samples. We investigate the effect of the total number of test samples (N) in EATA, where the fewer samples are sampled from the entire test set. From Table 9, EATA works well and consistently outperforms Tent counterpart, regardless of the number of test samples. Meanwhile, if there are many test samples, EATA would benefit more, *i.e.*, larger accuracy gain.

6. Conclusion

In this paper, we propose an efficient anti-forgetting test-time adaptation method, to improve the performance of pre-trained models on a potentially shifted test domain. Specifically, we devise a sample-efficient entropy minimization strategy that selectively performs test-time optimization with reliable and non-redundant samples. This improves the adaptation efficiency and meanwhile boosts the out-of-distribution performance. In addition, we introduce a Fisher-based anti-forgetting regularizer into test-time adaptation. With this loss, a model can be adapted continually without performance degradation on in-distribution test samples, making test-time adaptation more practical for real-world applications. Extensive experimental results on several benchmark datasets demonstrate the effectiveness of our proposed method.

Acknowledgments

References

Ashukha, A., Lyzhov, A., Molchanov, D., and Vetrov, D. P. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. In *International Conference on Learning Representations*, 2020.

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.

Berger, C., Paschali, M., Glocker, B., and Kamnitsas, K. Confidence-based out-of-distribution detection: A comparative study and analysis. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*, pp. 122–132. Springer, 2021.

Chen, P., Huang, D., He, D., Long, X., Zeng, R., Wen, S., Tan, M., and Gan, C. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI Conference on Artificial Intelligence*, volume 1, 2021.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Fan, D.-P., Zhou, T., Ji, G.-P., Zhou, Y., Chen, G., Fu, H., Shen, J., and Shao, L. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*, 39(8):2626–2637, 2020.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773, 2020.

Fleuret, F. et al. Test time adaptation through perturbation robustness. In *Advances in Neural Information Processing Systems Workshop*, 2021.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Guo, Y., Stutz, D., and Schiele, B. Improving corruption and adversarial robustness by enhancing weak subnets. *arXiv preprint arXiv:2201.12765*, 2022.

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

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8340–8349, 2021.

Iwasawa, Y. and Matsuo, Y. Test-time classifier adjustment module for model-agnostic domain generalization. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Koh, P. W., Sagawa, S., Marklund, H., Xie, S. M., Zhang, M., Balsubramani, A., Hu, W., Yasunaga, M., Phillips, R. L., Gao, I., et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664, 2021.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kundu, J. N., Venkat, N., Babu, R. V., et al. Universal source-free domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4544–4553, 2020.

Li, B., Wu, F., Lim, S.-N., Belongie, S., and Weinberger, K. Q. On feature normalization and data augmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12383–12392, 2021.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9641–9650, 2020.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

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

Lim, S., Kim, I., Kim, T., Kim, C., and Kim, S. Fast autoaugment. In *Advances in Neural Information Processing Systems*, volume 32, pp. 6665–6675, 2019.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Liu, Y., Kothari, P., van Delft, B., Bellot-Gurlet, B., Mordan, T., and Alahi, A. Ttt++: When does self-supervised test-time training fail or thrive? In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Madaan, D., Shin, J., and Hwang, S. J. Learning to generate noise for multi-attack robustness. In *International Conference on Machine Learning*, pp. 7279–7289, 2021.

Mittal, S., Galesso, S., and Brox, T. Essentials for class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3513–3522, June 2021.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

Niu, S., Wu, J., Zhang, Y., Guo, Y., Zhao, P., Huang, J., and Tan, M. Disturbance-immune weight sharing for neural architecture search. *Neural Networks*, 144:553–564, 2021.

Pei, Z., Cao, Z., Long, M., and Wang, J. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence*, 2018.

Qiu, Z., Zhang, Y., Lin, H., Niu, S., Liu, Y., Du, Q., and Tan, M. Source-free domain adaptation via avatar prototype generation and adaptation. In *International Joint Conference on Artificial Intelligence*, 2021.

Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T. P., and Wayne, G. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, 2019.

Rusak, E., Schott, L., Zimmermann, R. S., Bitterwolf, J., Bringmann, O., Bethge, M., and Brendel, W. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pp. 53–69, 2020.

Saito, K., Watanabe, K., Ushiku, Y., and Harada, T. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3723–3732, 2018.

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

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pp. 9229–9248, 2020.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

Wang, Q., Fink, O., Van Gool, L., and Dai, D. Continual test-time domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

Xu, G., Niu, S., Tan, M., Luo, Y., Du, Q., and Wu, Q. Towards accurate text-based image captioning with content diversity exploration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12637–12646, 2021.

Yao, H., Wang, Y., Li, S., Zhang, L., Liang, W., Zou, J., and Finn, C. Improving out-of-distribution robustness via selective augmentation. In *International Conference on Machine Learning*, 2022.

Zeng, R., Xu, H., Huang, W., Chen, P., Tan, M., and Gan, C. Dense regression network for video grounding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., and Gan, C. Graph convolutional module for temporal action localization in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Zhang, D., Ahuja, K., Xu, Y., Wang, Y., and Courville, A. Can subnetwork structure be the key to out-of-distribution generalization? In *International Conference on Machine Learning*, pp. 12356–12367, 2021a.

Zhang, M. M., Levine, S., and Finn, C. Memo: Test time robustness via adaptation and augmentation. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021b.

Zhang, Y., Niu, S., Qiu, Z., Wei, Y., Zhao, P., Yao, J., Huang, J., Wu, Q., and Tan, M. Covid-da: deep domain adaptation from typical pneumonia to covid-19. *arXiv preprint arXiv:2005.01577*, 2020a.

Zhang, Y., Wei, Y., Wu, Q., Zhao, P., Niu, S., Huang, J., and Tan, M. Collaborative unsupervised domain adaptation for medical image diagnosis. *IEEE Transactions on Image Processing*, 29:7834–7844, 2020b.

Zhang, Y., Hooi, B., Hong, L., and Feng, J. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv preprint arXiv:2107.09249*, 2021c.

Supplementary Materials for
"Efficient Test-Time Model Adaptation without Forgetting"

In the supplementary, we provide more implementation details and more experimental results of our EATA. We organize our supplementary as follows.

- In Section A, we provide more experimental details of our proposed EATA.

- In Section B, we show more experimental results to compare the out-of-distribution performance and efficiency with state-of-the-art methods on ImageNet-C with different corruption types and severity levels.

- In Section C, we give more experimental results to demonstrate the anti-forgetting ability of our EATA.

- In Section D, we provide more discussions on related training-time robustification studies.

A. More Implementation Details of EATA

A.1. More Details on Datasets

Following the settings of Tent (Wang et al., 2021) and MEMO (Zhang et al., 2021b), we conduct experiments on three benchmark datasets for out-of-distribution generalization, *i.e.*, CIFAR-10-C, ImageNet-C (Hendrycks & Dietterich, 2019) and ImageNet-R (Hendrycks et al., 2021).

CIFAR-10-C and **ImageNet-C** consist of corrupted versions of the validation images on CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009), respectively. The corruptions include 15 diverse types of 4 main categories (*i.e.*, noise, blur, weather, and digital). Each corruption type has 5 different levels of severity.

ImageNet-R contains 30,000 images with various artistic renditions of 200 ImageNet classes, which are primarily collected from Flickr and filtered by Amazon MTurk annotators.

A.2. More Experimental Protocols

Our EATA. Following TTT (Sun et al., 2020) and Tent (Wang et al., 2021), we use ResNet-26 and ResNet-50 for CIFAR-10 and ImageNet experiments, respectively. The models are trained on the original CIFAR-10 or ImageNet training set and then tested on clean or the aforementioned OOD test sets. For fair comparison, the parameters of ResNet-50 are directly obtained from the *torchvision*[1] library. ResNet-26 is trained via the official code of TTT by the same hyper-parameters, replacing the group norm with the batch norm, and removing the rotation head. For test time adaptation, we use SGD as the update rule, with a momentum of 0.9, batch size of 64, and learning rate of 0.005/0.00025 for CIFAR-10/ImageNet (following Tent and MEMO). The entropy constant E_0 in Eqn. (3) is set to $0.4 \times \ln C$, where C is number of task classes. The ϵ in Eqn. (6) is set to 0.4/0.05 for CIFAR-10/ImageNet. The trade-off parameter β in Eqn. (8) is set to 1/2,000 for CIFAR-10/ImageNet to make two losses have the similar magnitude. We use 2,000 samples to calculating $\omega(\theta_i)$ in Eqn. (9).

Compared Methods. For TTA (Ashukha et al., 2020), BN adaptation (Schneider et al., 2020) and MEMO (Zhang et al., 2021b), the hyper-parameters follow their original papers or MEMO. Specifically, the augmentation size of TTA (Ashukha et al., 2020) is set to 32 and 64 for CIFAR-10 and ImageNet, respectively. For BN adaptation (Schneider et al., 2020), both the batch size B and prior strength N are set to 256. The hyper-parameter settings of MEMO (Zhang et al., 2021b) can be found in their original paper. For Tent (Wang et al., 2021), we use SGD as the update rule with a momentum of 0.9. The batch size is 64 for both ImageNet and CIFAR-10 experiments. The learning rate is set to 0.00025 and 0.005 ImageNet and CIFAR-10, respectively. Note that the hyper-parameters of Tent are totally the same as our EATA for a fair comparison. For TTT (Sun et al., 2020), we strictly follow their original settings except for the augmentation size at test

---

[1]https://github.com/pytorch/vision

time for ImageNet experiments. According to TTT's implementation, the augmentation size is set to 64, which, however, is very time-consuming (*e.g.*, about 12 GPU hours on a single Tesla V100 GPU on ImageNet-C with a specific corruption type and severity level). In our implementation, we decrease this augmentation size to 20, which has only a slight performance difference compared with augmentation size 64. For example, the performances of TTT on ImageNet-C (Gaussian noise, severity level 5) with ResNet-18 are 26.2% (64) *vs.* 26.0% (20). We recommend the original papers of the above methods to readers for more implementation details.

B. More Results on Out-of-distribution Performance and Efficiency

In Table 12, we provide more results to compare our ETA and EATA with state-of-the-art methods on ImageNet-C with the severity level 1-4. Our ETA and EATA constantly outperform the state-of-the-art methods (*e.g.*, TTA, MEMO, and Tent) in most image corruption types of various severity levels. The performance gain mainly comes from the removal of the high-entropy test samples, since these samples may contribute unreliable and harmful gradients during test-time adaptation.

In Figure 5, we show the number of backward propagation of our ETA on ImageNet-C with different corruption types and severity levels. Across various corruption types, our ETA shows great superiority over existing methods in terms of adaptation efficiency. Compared with MEMO ($50,000\times64$) and Tent ($50,000$), our ETA only requires 31,741 backward passes (averaged over 15 corruption types) when the severity level is set to 3. The reason is that we exclude some unreliable and redundant test samples out of test-time optimization. In this case, we only need to perform backward computation on those remaining test samples, leading to improved efficiency.

Comparison with Tent using Different Learning Rates. In our sample-adaptive weight $S(\mathbf{x})$ in Eqn. (6), each test sample has a specific weight $S(\mathbf{x})$ and the value of $S(\mathbf{x})$ is always larger than 1. In this sense, training with sample-adaptive weight $S(\mathbf{x})$ indeed has the same effect as training with larger learning rates. Therefore, we compare our EATA with the baseline (Tent) using different learning rates. We increase the learning rate from 2.5×10^{-4} (which is the default of Tent) to 25.0×10^{-4} and report results in Table 10.

With the learning rate increasing from 2.5×10^{-4} to 10.0×10^{-4}, the error of Tent decreases from 45.3% to 43.9%, indicating that a larger learning rate may enhance the performance in some cases. However, when the learning rate becomes larger to 20.0×10^{-4}, the performance of Tent degrades. More critically, our EATA method outperforms Tent with varying learning rates. These results verify that simply enlarging the learning rate is not able to achieve competitive performance with our proposed sample-adaptive adaptation method, demonstrating our superiority.

Table 10. Comparison with Tent under different learning rates ($\times10^{-4}$) on ImageNet-C (Gaussian noise) regarding Error (%).

Severity	EATA (ours)	Tent (Wang et al., 2021)			
	$lr = 2.5$	$lr = 2.5$	$lr = 10.0$	$lr = 20.0$	$lr = 25.0$
Level 3	**42.6**	45.3	43.9	44.4	45.1
Level 5	**65.0**	71.6	72.2	83.6	87.1

EATA under Different Random Orders. Table 11 records EATA's performance (mean & stdev.) on randomly shuffled test samples with 10 different random seeds (from 2020 to 2029). From the results, EATA performs consistently across different random orders, showing the stability of EATA.

Table 11. The mean and stdev. of corruption accuracy (%) of EATA over 10 random orders, on ImageNet-C (level 5) with ResNet-50.

Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Avg.
$34.9_{\pm0.2}$	$36.9_{\pm0.1}$	$35.8_{\pm0.2}$	$33.6_{\pm0.3}$	$33.3_{\pm0.2}$	$47.2_{\pm0.3}$	$52.7_{\pm0.1}$	$35.8_{\pm0.2}$

C. More Results on Prevent Forgetting

In this section, we provide more results to demonstrate the effectiveness of our EATA in preventing forgetting. We report the comparison results of EATA (lifelong) *vs.* Tent (lifelong) and EATA *vs.* Tent in Figures 6 and 7, respectively. In the lifelong adaptation scenario, Tent suffers more severe ID performance degradation than that of reset adaptation (*i.e.*, Figure 7), showing that the more optimization steps, the more severe forgetting. Moreover, with the increase of the severity level, the

ID clean accuracy degradation of Tent increases accordingly. This result indicates that the OOD adaptation with more severe distribution shifts will result in more severe forgetting. In contrast, our methods achieve higher OOD corruption accuracy and meanwhile maintain the ID clean accuracy (competitive to the original accuracy that tested before any OOD adaptation) in both two adaptation scenarios (reset and lifelong). These results are consistent with that in the main paper and further demonstrate the effectiveness of our proposed anti-forgetting weight regularization.

D. More Discussions on Related Training-Time Robustification

To defend against distribution shifts, many prior studies seek to enlarge the training data distribution to enable it to cover the possible shift that might be encountered at test time, such as adversarial training strategies (Wong et al., 2020; Rusak et al., 2020; Madaan et al., 2021), various data augmentation techniques (Lim et al., 2019; Hendrycks et al., 2020; Li et al., 2021; Hendrycks et al., 2021; Yao et al., 2022) and searching/enhancing sub-networks of a deep model (Zhang et al., 2021a; Guo et al., 2022). However, it is hard to anticipate all possible test shifts at training time. In contrast, we seek to conquer this test distribution shift by directly learning from test data.

Table 12. Comparisons with state-of-the-art methods on ImageNet-C with the severity levels 1-4 regarding **Error (%)**. "GN" and "BN" denote group normalization and batch normalization, respectively. "JT" denotes the model is jointly trained via supervised cross-entropy loss and rotation prediction loss. The **bold** number indicates the best result and the underlined number indicates the second best result.

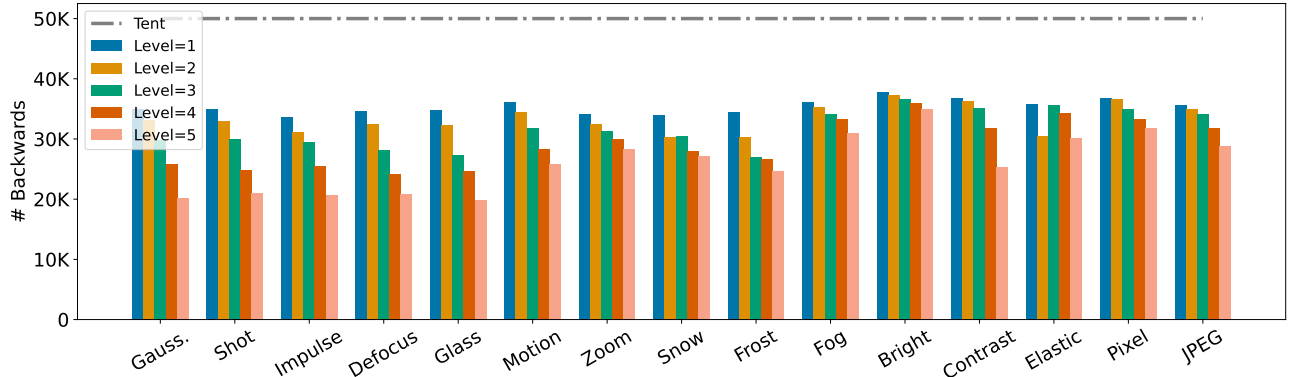|	Noise			Blur				Weather				Digital				Average	
Severity level=1	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	#Forwards	#Backwards
R-50 (GN)+JT	40.9	43.4	52.4	47.4	48.2	39.9	56.2	48.4	45.5	45.2	31.6	35.5	36.7	36.7	39.2	50,000	0
• TTT	37.5	37.4	39.4	41.8	39.7	37.2	43.9	41.7	40.5	36.3	31.0	32.9	35.0	33.8	36.2	50,000×21	50,000×20
R-50 (BN)	40.6	42.8	52.0	40.6	46.0	35.3	47.5	45.4	38.7	38.2	26.0	35.1	33.4	35.9	33.8	50,000	0
• TTA	40.6	42.6	52.5	44.4	46.6	38.0	45.6	48.0	42.0	40.6	28.8	35.6	33.9	37.7	36.8	50,000×64	0
• BN adaptation	34.6	36.1	41.2	35.7	35.2	30.8	37.6	38.2	35.2	31.3	**25.4**	28.5	30.8	28.7	30.5	50,000	0
• MEMO	36.9	39.5	46.3	38.2	41.1	32.8	42.7	40.4	36.8	35.3	25.8	31.3	31.2	32.4	32.9	50,000×65	50,000×64
• Tent	32.2	32.7	36.2	33.8	32.8	29.7	34.6	35.1	33.6	29.8	_25.6_	28.0	30.1	28.0	29.8	50,000	50,000
• Tent (episodic)	36.0	37.8	41.7	38.8	38.1	32.0	39.3	40.5	37.3	32.6	26.5	29.8	31.7	29.9	32.1	50,000×2	50,000
• ETA (ours)	31.7	_31.8_	_34.7_	32.9	32.2	29.6	34.1	**33.6**	33.3	29.5	26.0	28.2	30.3	28.1	29.9	50,000	35,379
• EATA (ours)	**31.5**	_31.8_	34.9	33.0	_32.1_	_29.2_	**33.8**	**33.6**	33.0	_29.4_	25.7	**27.7**	29.9	27.8	29.6	50,000	34,898
• EATA (lifelong)	**31.5**	**31.7**	34.6	32.9	32.0	**29.1**	33.8	**33.6**	33.0	29.3	25.8	**27.6**	29.9	27.8	29.6	50,000	36,675
Severity level=2	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	#Forwards	#Backwards
R-50 (GN)+JT	50.1	55.0	60.0	55.8	61.6	49.9	65.5	68.9	63.1	52.0	33.5	38.6	58.0	39.4	42.8	50,000	0
• TTT	42.0	42.5	44.3	47.7	53.3	42.7	48.2	50.4	56.6	38.2	31.8	34.3	49.4	34.7	38.3	50,000×21	50,000×20
R-50 (BN)	53.8	57.9	64.2	48.0	59.6	45.7	57.4	68.1	55.9	44.1	27.6	41.6	55.2	37.5	37.5	50,000	0
• TTA	52.9	56.9	62.4	53.0	60.4	49.3	54.4	71.5	60.6	46.9	30.7	40.0	53.8	41.1	40.3	50,000×64	0
• BN adaptation	42.3	45.6	49.7	43.0	44.5	37.4	44.1	53.1	47.6	34.1	**26.6**	30.6	47.2	29.7	33.8	50,000	0
• MEMO	47.1	51.7	55.9	44.8	53.4	41.3	51.8	58.6	51.6	40.2	27.2	35.5	51.0	33.6	36.2	50,000×65	50,000×64
• Tent	37.2	38.5	41.8	39.5	39.4	34.0	39.2	44.8	43.6	31.6	**26.6**	29.7	30.0	32.6	32.1	50,000	50,000
• Tent (episodic)	44.0	47.2	50.5	48.0	49.3	39.2	46.4	55.0	50.7	35.5	27.6	32.3	49.1	32.2	36.1	50,000×2	50,000
• ETA (ours)	**35.8**	_36.5_	_39.5_	37.7	_37.8_	_33.1_	37.7	**41.4**	_41.7_	30.9	27.0	29.2	43.0	28.6	31.8	50,000	33,363
• EATA (ours)	35.9	_36.5_	39.6	37.9	_37.8_	_33.1_	_37.4_	41.7	_41.7_	**30.7**	26.7	**29.1**	42.6	28.4	31.4	50,000	32,754
• EATA (lifelong)	35.9	**36.2**	**39.2**	37.7	37.6	**33.0**	37.3	_41.6_	41.6	**30.8**	26.6	**29.1**	42.6	_28.5_	31.4	50,000	34,922
Severity level=3	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	#Forwards	#Backwards
R-50 (GN)+JT	64.4	69.3	66.7	71.5	83.2	65.8	71.4	65.5	73.9	61.3	36.7	45.2	44.1	50.2	45.5	50,000	0
• TTT	48.5	48.9	48.3	57.7	67.1	50.7	50.7	51.2	70.3	41.0	33.3	37.6	36.3	38.4	39.9	50,000×21	50,000×20
R-50 (BN)	72.4	75.0	74.9	62.0	83.1	62.3	64.8	64.8	67.9	53.4	30.4	54.0	44.4	53.8	40.7	50,000	0
• TTA	70.4	72.7	70.0	68.7	83.7	66.2	62.0	67.9	72.7	56.3	33.8	47.7	47.8	51.8	43.4	50,000×64	0
• BN adaptation	54.7	57.2	56.6	57.8	63.6	48.9	48.8	52.2	57.5	38.2	28.3	35.4	33.1	36.0	36.6	50,000	0
• MEMO	62.5	65.7	63.3	58.8	76.7	55.9	58.9	55.6	62.6	47.9	29.5	44.0	41.3	45.0	39.2	50,000×65	50,000×64
• Tent	45.3	45.9	46.6	51.1	53.9	41.2	42.4	44.4	51.5	34.2	27.9	32.9	30.7	32.6	34.3	50,000	50,000
• Tent (episodic)	56.6	58.8	57.5	64.9	69.5	51.5	51.0	54.4	60.3	39.9	29.5	37.7	35.3	37.5	39.1	50,000×2	50,000
• ETA (ours)	**42.4**	**42.4**	_43.3_	_47.3_	**49.6**	38.9	40.5	**41.2**	**48.5**	33.1	28.1	32.0	30.5	**31.7**	33.6	50,000	31,741
• EATA (ours)	_42.6_	42.9	43.7	47.4	49.8	_38.8_	_40.4_	_41.6_	_48.7_	**32.7**	**27.7**	**31.7**	30.0	31.8	33.3	50,000	31,068
• EATA (lifelong)	_42.6_	42.4	43.2	47.2	**49.6**	38.7	40.2	_41.6_	48.8	32.9	27.8	**31.7**	_30.1_	**31.7**	_33.4_	50,000	33,469
Severity level=4	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	#Forwards	#Backwards
R-50 (GN)+JT	81.4	87.6	82.2	81.8	87.8	80.2	77.2	76.8	75.9	66.9	41.8	64.2	54.8	68.0	54.7	50,000	0
• TTT	57.5	58.7	57.1	64.8	84.9	60.5	54.4	57.3	73.1	43.1	35.5	46.0	39.6	44.5	45.1	50,000×21	50,000×20
R-50 (BN)	89.0	92.1	91.8	73.5	87.2	78.1	71.6	75.9	70.1	59.6	34.9	79.5	57.8	71.1	52.5	50,000	0
• TTA	85.9	89.5	86.1	79.8	88.4	80.5	67.9	79.3	74.4	62.9	39.3	65.8	61.0	66.2	53.0	50,000×64	0
• BN adaptation	69.3	74.1	71.2	70.3	70.5	63.0	54.8	62.0	58.9	41.5	30.8	50.0	38.7	44.7	46.0	50,000	0
• MEMO	78.5	83.3	78.8	71.0	82.2	71.2	67.6	65.9	64.5	53.3	33.4	66.7	52.8	58.6	48.7	50,000×65	50,000×64
• Tent	56.0	59.4	57.3	61.7	60.6	50.9	46.5	51.4	53.1	36.5	30.0	43.3	34.0	37.9	39.6	50,000	50,000
• Tent (episodic)	70.7	75.5	72.0	77.3	76.5	66.3	57.2	64.1	61.8	43.1	32.0	56.4	41.4	46.5	48.9	50,000×2	50,000
• ETA (ours)	**51.5**	_53.4_	_52.5_	57.1	_55.4_	46.6	_43.7_	_46.9_	**49.7**	35.1	_29.8_	39.4	33.2	**36.0**	38.2	50,000	29,240
• EATA (ours)	_52.3_	54.2	53.0	_57.0_	55.5	**46.6**	_43.7_	**46.8**	49.9	**34.7**	**29.7**	_39.8_	**33.1**	_36.4_	_38.3_	50,000	28,423
• EATA (lifelong)	_52.3_	**53.0**	**52.2**	56.4	55.1	**46.6**	**43.4**	47.2	**49.7**	_34.8_	_29.8_	40.0	**33.1**	36.5	38.5	50,000	31,141

Figure 5. Comparison between ETA and Tent in terms of the number of backward propagation on ImageNet-C with different corruption types and severity levels.
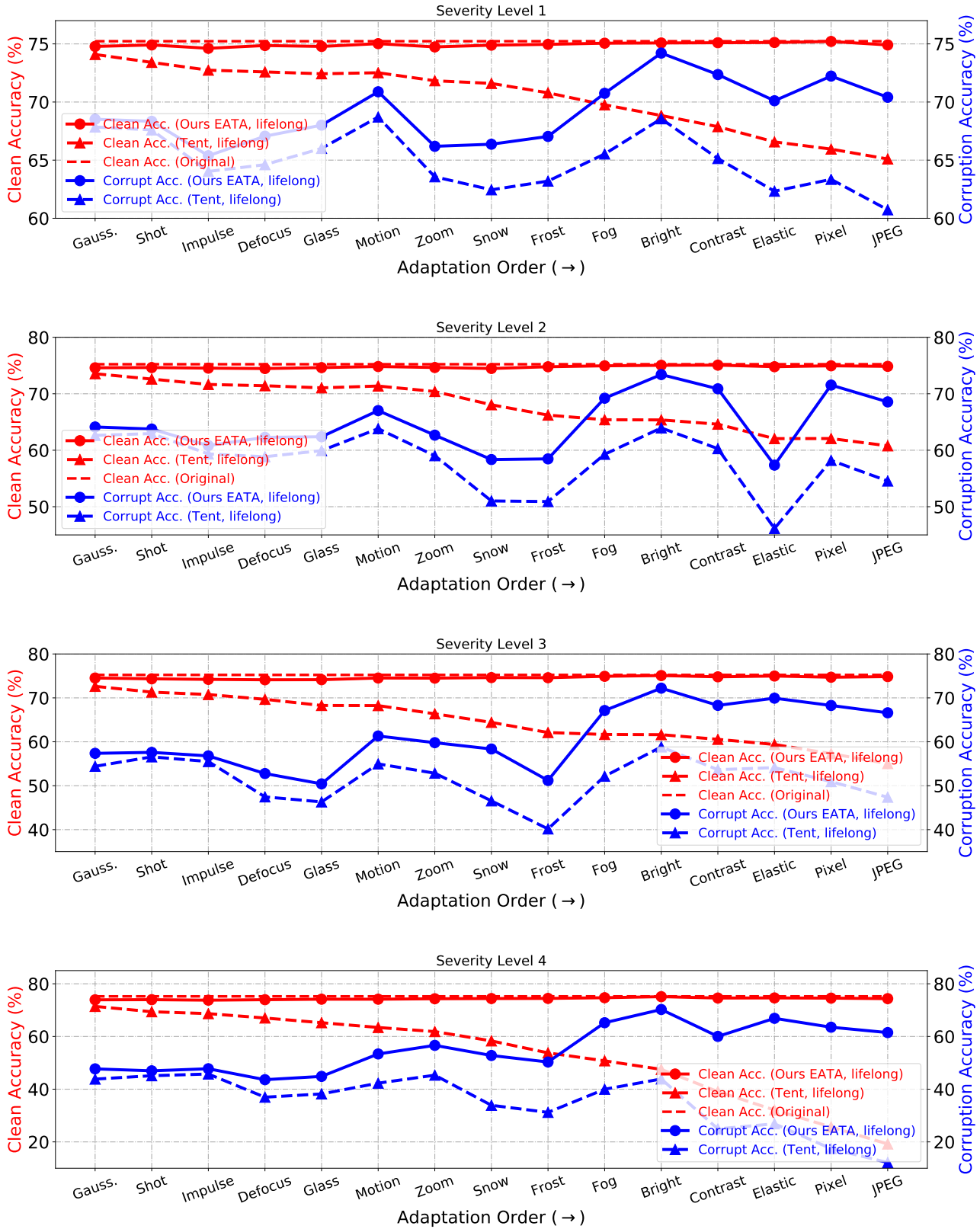
Figure 6. Comparison of prevent forgetting on ImageNet-C (severity levels 1-4) with ResNet-50. We record the OOD corruption accuracy on each corrupted test set and the associated ID clean accuracy (after OOD adaptation). The model performs lifelong adaptation, in which the model parameters will never be reset.
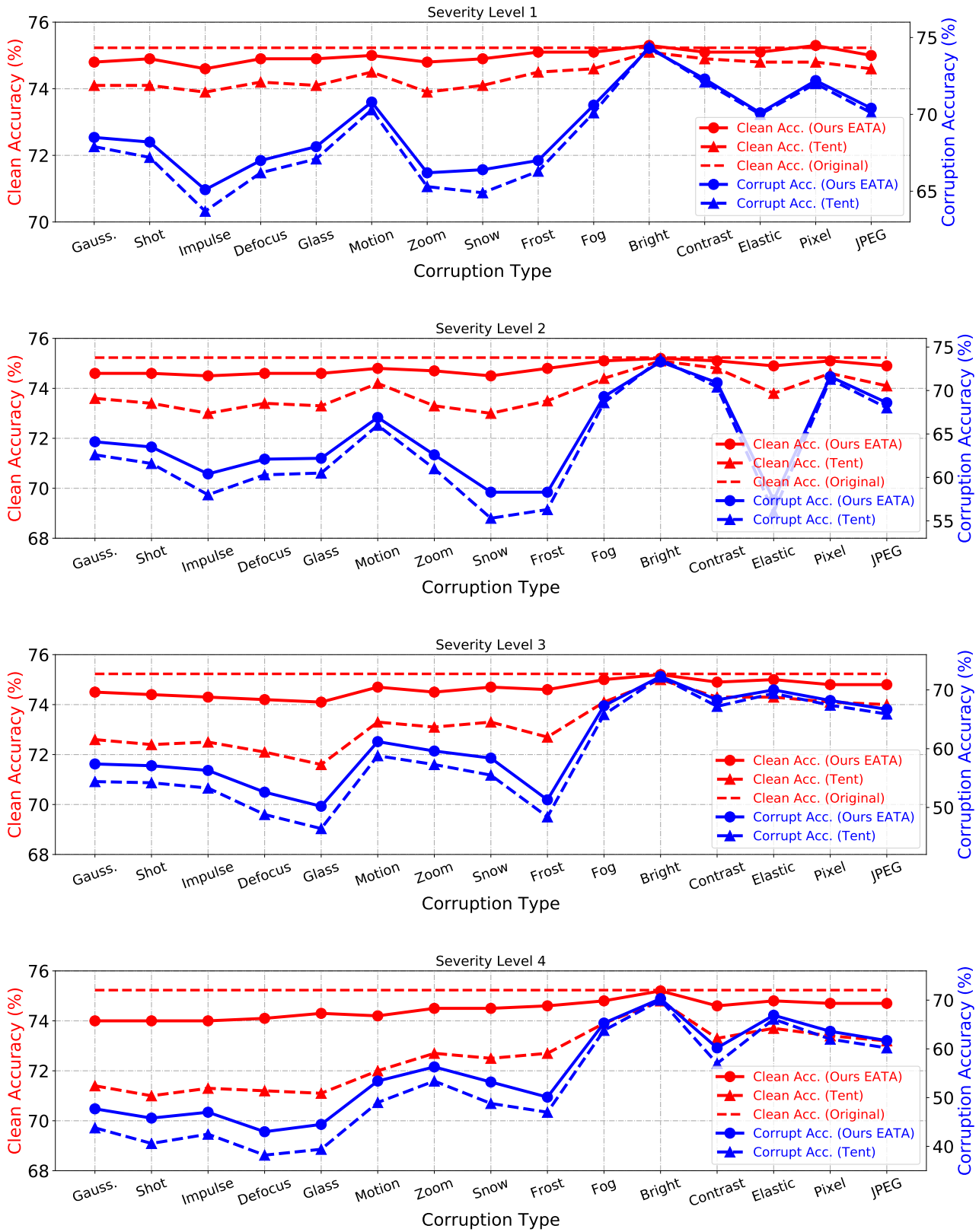
Figure 7. Comparisons of prevent forgetting on ImageNet-C (severity levels 1-4) with ResNet-50. We record the OOD corruption accuracy on each corrupted test set and the associated ID clean accuracy (after OOD adaptation). The model parameters of both Tent and our EATA are reset before adapting to a new corruption type.