REALM: Robust Entropy Adaptive Loss Minimization for Improved Single-Sample Test-Time Adaptation

Skyler Seto, Barry-John Theobald, Federico Danieli, Navdeep Jaitly, Dan Busbridge
Apple
{sseto,barryjohn_theobald, f_danieli, njaitly, dbusbridge}@apple.com

Abstract

Fully-test-time adaptation (F-TTA) can mitigate performance loss due to distribution shifts between train and test data (1) without access to the training data, and (2) without knowledge of the model training procedure. In online F-TTA, a pre-trained model is adapted using a stream of test samples by minimizing a self-supervised objective, such as entropy minimization. However, models adapted with online using entropy minimization, are unstable especially in single sample settings, leading to degenerate solutions, and limiting the adoption of TTA inference strategies. Prior works identify noisy, or unreliable, samples as a cause of failure in online F-TTA. One solution is to ignore these samples, which can lead to bias in the update procedure, slow adaptation, and poor generalization. In this work, we present a general framework for improving robustness of F-TTA to these noisy samples, inspired by self-paced learning and robust loss functions. Our proposed approach, Robust Entropy Adaptive Loss Minimization (REALM), achieves better adaptation accuracy than previous approaches throughout the adaptation process on corruptions of CIFAR-10 and ImageNet-1K, demonstrating its effectiveness.

1. Introduction

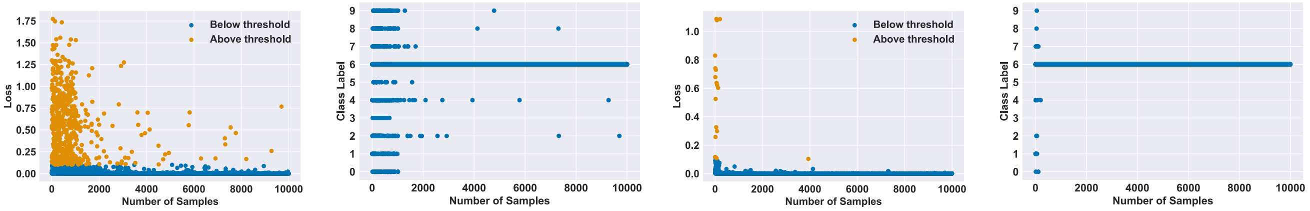
Deep Neural Networks (DNNs) can achieve excellent performance when evaluated on data from the same distribution used to train the model. However, after model deployment, natural variations, sensor degradation, or changes in the environment can cause test samples to appear different from the samples used to train the model. Such distribution shifts, may significantly worsen performance [15].

Test-time adaptation (TTA) is a strategy used to counter distribution shifts during online evaluation/model deployment. TTA updates the model parameters within the inference procedure through self-supervision. In the *Fully-TTA* (F-TTA) setting, the goal is to adapt an arbitrary pre-trained model on test data without access to the original training data (also called source-free), without supervision, and without access to changing the way the model was trained [44]. F-TTA is an important solution for tackling source-free distribution shifts when: (1) models are deployed and source data are not available, e.g., for privacy concerns, (2) it may be prohibitively expensive to re-train the models, and (3) data from the unseen distribution may not be available immediately, and it is infeasible to wait for enough data to annotate and train with supervision.

A common paradigm in F-TTA is training models to minimize the entropy of the predictions [44], which is typically done in two ways: (1) episodic: where models are updated and reset after each batch of samples, and (2) online: where weights are not reset after each test sample allowing for update accumulation [29]. While several papers have examined F-TTA [11, 13, 20, 22, 37–40, 44, 49] for online adaptation, many methods are sensitive to known issues: (1) batch normalization running statistics updating on small number of samples from the new distribution [39], (2) noisy samples with high entropy leading to unstable model updates [38, 39], and (3) hyper-parameter sensitivity causing models to shift too much from the source model [50]. Furthermore, online F-TTA has been shown to yield worse performance as the amount of adaptation data increases, and to be prone to over-fitting leading to degenerate solutions [29, 39]. Additionally, little is known about online F-TTA performance in extreme scenarios with a limited number of adaptation steps, and limited amount of data per adaptation step (i.e., batch size of one for real-world inference deployment[1]).

In this work, we examine the reliability of current approaches for online F-TTA with limited data *and* limited adaptation step settings. It is important to have *both* of these conditions, as restricting to *only* a limited number of adaptions does not constrain optimization, providing the

[1] A natural example might be an individual taking picture(s) on their phone on a rainy day. A phone app providing object recognition capabilities would be expected to classify on the new distribution given only a few samples, which are processed individually.



(a) TENT Loss

(b) TENT Predicted Class

(c) MEMO Loss

(d) MEMO Predicted Class

Figure 1. Loss and predicted class labels for both TENT and MEMO on the CIFAR-10 test set under gaussian noise corruption at severity 5. Both adaptation strategies are applied to a ResNet-26 with group norm layers using standard hyperparameters from [44] and [49], but in the online adaptation setting and a batch size of 1. (a), (c): Certain samples have high loss at th start of training. (b), (d): Model predicted classes quickly become the same across all inputs.

stochastic differential equation (SDE) approximation of the optimization is valid[2] [32]. Equivalently, restricting to a small number of samples per adaption step does not constrain the optimization, whereas restricting to a small number of total samples does.

Our main contributions are (1) We highlight that with batch size of one[3], online F-TTA can fail to adapt to the target distribution and instead quickly moves toward a degenerate solution (predicting the same label for all samples) due to high loss sample. (2) We study procedures which perform stable online F-TTA by skipping samples according to a reliability criteria [38,39], which identify that unreliable noisy samples are a cause of unstable adaptation, and characterize their objective as part of a broader objective relating to Self-Paced Learning (SPL) [10, 27]. (3) We propose a general variant of the SPL framework for online TTA called Robust Entropy Adaptive Loss Minimization (REALM) that updates using samples scaled by a robust loss function.

Empirically, REALM yields better performance at early stages of adaptation (few adaptation steps/limited samples), and leads to better performance over the full test set compared to related entropy minimization methods.

2. Reliable Test Time Adaptation

In this section, we give a brief overview of test time adaptation, show that strategies based on entropy minimization fail to adapt to the distribution shift due to noisy samples, and define prior work that skip samples for stable TTA.

2.1. Overview of Test Time Adaptation

Let $f(\cdot\,; \theta)$ be a model trained on a training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i) \sim P_{\text{train}}\}_{i=1}^N$. The goal of TTA is to improve the performance of $f(\cdot\,;\theta)$ on the evaluation of a test distribution P_{test}, where $P_{\text{train}} \neq P_{\text{test}}$, without access to how $f(\cdot\,;\theta)$ was trained, and without access to $\mathcal{D}_{\text{train}}$. In practice when optimizing over the test set, we have access to the batch of samples x without corresponding labels y, that is $\mathcal{D}_{\text{test}} = \{(x_i)_{i=1}^M \sim Q\}$. In TTA, the model parameters θ are adapted by batchwise minimizing over the test data:

$$\theta^* = \min_{\theta} \mathcal{L}_{\text{SSL}}(\theta; \mathcal{D}_{\text{test}}), \tag{1}$$

where $\mathcal{L}_{\text{SSL}}$ is some self-supervised objective, such as minimizing entropy [44], or marginal entropy [49]. The goal of TTA is to adapt the model by optimizing the above objective in an online inference setting, where batches of data are streamed to the model, and predictions are made on-the-fly. This setting is challenging, especially when the amount of data at each step is limited. The model is expected to perform well immediately in the adaptation phase, yet adaption might involve only a small number of updates to keep inference time efficient, and data might not be stored (e.g., for sensitive data with privacy considerations, or limited storage devices). *Note that in the online setting no termination is necessary, however in episodic settings, the model can be reset after new or no data appears.*

2.2. Limitations of Test Time Adaptation

In online adaptation scenarios with limited batch sizes, we observe that a model may quickly overfit to the self-supervised objective $\mathcal{L}_{\text{SSL}}$, also known as entropy collapse, resulting in performance degradation as the model maximizes its confidence by predicting the same class label. Figure 1 illustrates this effect with two baseline TTA strategies: TENT [44] and MEMO [49], which minimizes entropy irrespective of the sample. When using each test sample to adapt the model, some samples seen early in adaptation have high loss (Fig. 1), and when these samples are used to adapt the model, the model learns to always predict the same class label. Concurrent work, SAR, identified a similar phenomena with batched TENT through comparison of

[2] It is likely that the SDE approximation holds in the TTA setting as TTA is performed in the low-LR, small batch size regime, which corresponds to low discretization [32].

[3] In this work, we refer to this setting as single sample TTA. While we adapt using the entire evaluation set, seeing samples individually still presents a challenge as the model update over a single sample can be very noisy and unreliable.

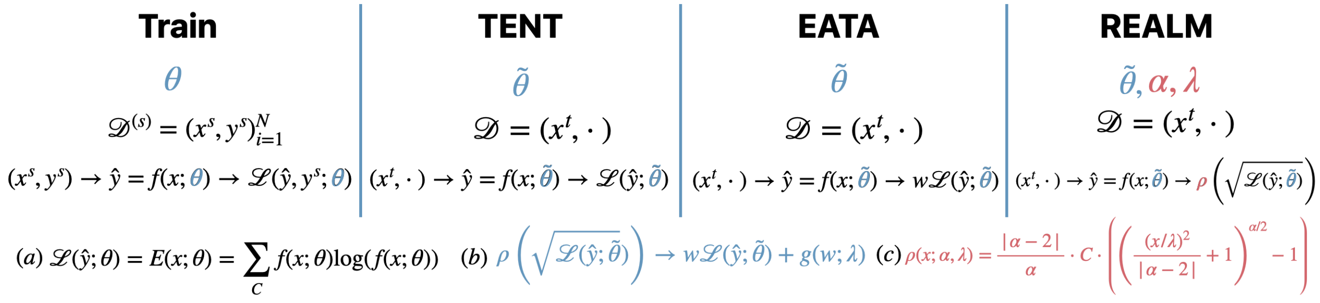

Train
TENT
EATA
REALM
θ
$\tilde{\theta}$
$\tilde{\theta}$
$\tilde{\theta}, \alpha, \lambda$
$\mathscr{D}^{(s)} = (x^s, y^s)_{i=1}^N$
$\mathscr{D} = (x^t, \cdot)$
$\mathscr{D} = (x^t, \cdot)$
$\mathscr{D} = (x^t, \cdot)$
$(x^s, y^s) \rightarrow \hat{y} = f(x; \theta) \rightarrow \mathscr{L}(\hat{y}, y^s; \theta)$
$(x^t, \cdot) \rightarrow \hat{y} = f(x; \tilde{\theta}) \rightarrow \mathscr{L}(\hat{y}; \tilde{\theta})$
$(x^t, \cdot) \rightarrow \hat{y} = f(x; \tilde{\theta}) \rightarrow w\mathscr{L}(\hat{y}; \tilde{\theta})$
$(x^t, \cdot) \rightarrow \hat{y} = f(x; \tilde{\theta}) \rightarrow \rho\left(\sqrt{\mathscr{L}(\hat{y}; \tilde{\theta})}\right)$
$(a)\ \mathscr{L}(\hat{y}; \theta) = E(x; \theta) = \sum_C f(x; \theta) \log(f(x; \theta))$
$(b)\ \rho\left(\sqrt{\mathscr{L}(\hat{y}; \tilde{\theta})}\right) \rightarrow w\mathscr{L}(\hat{y}; \tilde{\theta}) + g(w; \lambda)$
$(c)\ \rho(x; \alpha, \lambda) = \frac{|\alpha - 2|}{\alpha} \cdot C \cdot \left(\left(\frac{(x/\lambda)^2}{|\alpha - 2|} + 1\right)^{\alpha/2} - 1\right)$

Figure 2. An overview of REALM and similar methods. For TTA, the model $f(\cdot\,; \theta)$ training procedure stays fixed, while inference changes. (a) TENT proposes batch-wise entropy minimization of all samples, while EATA minimizes the entropy of only samples with low entropy according to the weight $w = S(x)$. We formalize optimization procedure of EATA as (b) an instance of Self-Paced Learning. Our algorithm REALM, minimizes (c) a robust loss function ρ of the entropy to stabilize online adaptation against outliers. We show that our proposed robust loss function generalizes (b).

the entropy and gradient norms on ImageNet with gaussian noise corruption at severity 5 [38].

To mitigate entropy collapse, recent works propose reweighting the entropy objective with an indicator function on the sample entropy that results in skipping updates for high entropy (i.e., *unreliable*) samples [38, 39]. The EATA approach [38] introduces the objective:

$$\min_{\theta} S(x)\, \mathcal{L}(\theta; x). \tag{2}$$

The particular weight $S(x)$ is written in two parts:

$$S(x) = S_{\text{ent}}(x) \cdot S_{\text{div}}(x), \tag{3}$$

$$S_{\text{ent}}(x) = \frac{1}{\exp\left[\mathcal{L}(\theta) - \lambda\right]} \cdot \mathbb{1}\left\{\mathcal{L} < \lambda\right\}, \tag{4}$$

for some pre-defined threshold on the entropy, λ. Such a procedure has also been used for building robust optimizers [41]. S_{div} is defined as:

$$S_{\text{div}} = \mathbb{1}\left\{\cos\left(f(x; \theta), m_{t-1}\right) \ < d\right\}, \tag{5}$$

where m_{t-1} is an Exponential Moving Average (EMA) of the predictions from prior updates[4].

However, as EATA is an online procedure, when samples are seen only once, some samples may never be used to update the model, especially with a batch size of one, which can lead to class-imbalance and potential bias in the update procedure. Further, this process results in few updates at the start of testing, which results in slow adaptation to the shifted distribution, and a dependency on the ordering of samples during adaptation. Concurrent to this work, SAR [39] extends EATA by also modifying the optimization procedure with Sharpness Aware Minimization (SAM) [12], which encourages the model to move towards flat entropy. An overview of F-TTA approaches such as Tent and EATA[5] are shown in Fig. 2. The primary differenec between our approach REALM and prior works is that we modify the entropy minimization objective using a robust loss function ρ, a generalization of the weighting approach in EATA, and Tent which induces entropy collapse.

3. Related Work

The goal of this work is to adapt a pre-trained model trained on (source) data using only unlabeled data from a new (target) distribution at test time. In the online setting, the data are assumed to arrive in a streaming fashion, and can be used only once to update the model. This is in contrast to similar fields, such as *domain adaptation* [26], which trains a model with data from the training distribution (source) and unlabeled data from the test distribution (target), and *domain generalization* [51], which trains a model on multiple domains to generalize to new domains at test time. Although there are source-free methods within the *domain adaptation* literature [28, 30, 34], methods that modify the training procedure for better TTA such as Test-time training [9, 36, 43], and methods that carry an additional model, such as a diffusion model trained on source data [13], they assume access to a large amount of data from the source or target distribution for potentially multiple epochs of training, and are outside the scope of this work. For our work, we focus on online TTA methods that minimize an entropy-based objective.

One of the earliest works in TTA investigates adaptation of a speaker-independent acoustic model to new speakers at

[4] Note that S_{div} is operationalized as a `torch.where()` and a stop gradient is applied such that no update is done even though S is a function of θ. Thus, we can simply think of $S(x)$ as a function of the sample only.

[5] We refer to the online F-TTA setting in this work as just TTA as all comparisons made all primarily operate under the online Fully TTA setting. We specify when comparing other methods and in the related work when such methods operate under a different setting.

test-time [46]. Numerous works have since proposed different approaches for TTA in vision [31, 36, 40, 43, 44, 49], and speech [23, 35]. Recent TTA strategies primarily focus on small updates to the model, typically only adapting the normalization layers of the network (i.e. batch normalization [19], group normalization [47], and layer normalization [1]) as these layers have a small number of parameters relative to the full network, and are impacted heavily by covariate shifts [3, 40]. AdaBN [3, 31] suggests updating the statistics in BN layers according to the new distribution, and [40] suggests a rolling update of the normalziation layer statistics. Other approaches also use variants of the batch (such as augmentations) to perform normalization statistic updates [22]. Other methods update the parameters (momentum and scale) of the normalziation layers including [38,39,44]. Still, other methods selectively update parts of the full network [29,39] or the entire network [49], however apriori it is hard to know what parts of the network should be updated for an unknown distribution shift.

Many normalization layer update methods [3,31,38,44] rely on a large batch of samples (more than 64) to perform adaptation, and are thus unsuitable for online TTA, especially in the single instance, or few sample regime. A recent investigation into online TTA, TENT [44], minimizes the entropy of a given batch of data and continues on new batches of data. TTT also maintains an online version, but still presupposes training with a different objective [43].

Following on, numerous approaches have proposed unsupervised entropy-based optimizations including [21] for bayesian domain adaptation, MEMO and TTA-PR [11, 49], which optimize average or marginal entropy over multiple augmentations. Recent works additionally investigate the instability of online TTA with entropy minimization demonstrating catastrophic drop in performance in instance-based online TTA, and poor performance from adapting on unreliable samples. In particular, EATA [38] suggests skipping updates on samples that have high entropy, and SAR [39] concurrent to our work notes that updating on these samples leads to overfitting the unsupervised objective, resulting in a model which always predicts the same class. The SAR procedure uses the same weight objective as in EATA, but also uses the SAM optimizer [12]. Other works demonstrate that limiting the parts of the model that are updated can lead to more stable online TTA [29]. In this work, we critically examine these approaches for improving stability in online TTA with entropy minimization, highlighting their shortcomings and proposing an improved, general framework encompassing these approaches.

Beyond entropy minimization, pseudo-label approaches perform online TTA [6, 24], and many other attempts at attaining stability from continual learning [7] including anti-forgetting regularization, and consistency regularization constrain the model parameters to be close to the source model, thus improving stability [25,38,45]. However, these approaches have a different goal aimed at reducing forgetting source distribution, and not in directly stabilizing and improving adaptation to a target domain via entropy minimization objectives. Many approaches concurrent to this work also rely on memory banks [48], storing original model weights [45], or copies of the model [42]. For a comprehensive survey on these approaches, and other TTA approaches see [33].

4. Robust Entropy Adaptive Loss Minimization (REALM)

4.1. Connecting EATA to Self-Paced Learning and Robust Loss Functions

Consider the empirical risk minimization objective:

$$\theta^* = \arg\min_\theta \mathbb{E}\left[\mathcal{L}_S(\theta; x)\right] = \arg\min_\theta \sum_{i=1}^{N} \mathcal{L}_S(\theta; x_i). \quad (6)$$

Given the weight S_{ent} from (4), we rewrite the reweighted objective (2) through a connection to the SPL literature [10, 27]. In SPL, the aim is to solve the joint optimization problem in w and θ:

$$\begin{aligned} w^*, \theta^* &= \arg\min_{w,\theta} \mathbb{E}\left[w(x)\mathcal{L}(\theta; x) + g(w; \lambda)\right], \\ &= \arg\min_{w,\theta} \sum_{i=1}^{N} [w(x_i)\mathcal{L}(\theta; x_i) + g(w(x_i); \lambda)], \end{aligned} \quad (7)$$

where $w(x_i) \in [0, 1]$ is a weight for the loss controlling the importance of the sample for learning, and $g(\cdot)$ is a regularizer on w controlling the pace of learning.

A typical procedure for solving Eq. (7) is an alternating iterative procedure, where one first solves for the optimal weights w^* holding θ fixed, then the model parameters θ^* while holding w fixed. We write EATA [38] as a SPL objective by defining $g(w; \lambda) = -\lambda \|w\|_1$, which yields the min-min objective:

$$w^*, \theta^* = \arg\min_{w,\theta} \mathbb{E}\left[w(x)\mathcal{L}(\theta; x) - \lambda \|w(x)\|_1\right]. \quad (8)$$

For this choice of g, its closed form solution for w is $S_{\text{ent}}(x)$, and optimization of θ is same procedure as (2) up to some constants in θ, which do not influence optimization [27].

We further note that for certain classes of regularizer, the SPL optimization can be written in terms of a robust loss function $\rho(x)$. That is, optimization of the form $\min_\theta \sum_{i=1}^{N} \rho(\mathcal{L}(\theta; x_i))$ as is done in SPL with implicit[6] regularization [10].

[6] Implicit means that one need not have a closed-form expression for g. Nonetheless, our choice of ρ does have a closed form g.

For hard-thresholding, the corresponding robust loss function is the Talwar function [17], and for EATA, the corresponding "robust" loss function is

$$\rho_{\text{EATA}}(x;\lambda) = \begin{cases} x & x \leq \lambda \\ \lambda & \text{otherwise}, \end{cases} \tag{9}$$

where λ is the loss threshold and is x-independent. In summary, we have shown that EATA is optimization of a weighted Talwar loss, or an SPL objective with an L1 norm on the weights.

4.2. Robust Entropy Adaptive Loss Minimization

One may intuitively think the corresponding loss function and weighting function need to correspond to piecewise penalization according to the relationship between the loss threshold λ, similar to a robust loss like the Huber loss [18]. However, there are many suitable "robust" loss functions that are not piecewise, for example the Welsch loss function [8], which yields a similar tapering on the loss, and the Cauchy loss function [5].

In this work, we suggest adaptation with a general robust loss function which interpolates many robust loss functions used in literature. To our knowledge this is the first instance of such a function applied on the entropy objective, and for TTA. The form for the adaptive loss function is written as:

$$\rho(x;\alpha,\lambda) = \frac{|\alpha-2|}{\alpha}\cdot C\cdot\left[\left(\frac{(x/\lambda)}{|\alpha-2|}+1\right)^{\alpha/2}-1\right], \tag{10}$$

for $\alpha \in (0,2]$, and has been adapted[7] from [2] for entropy minimization. Optimization of Eq. (10) also has the benefit of parameterizing both the shape of the loss (in terms of α and the threshold λ in terms of the scale of the loss). Thus, we suggest optimization of the general robust function of the entropy for better TTA. This yields a far simpler procedure for stable TTA while still being robust to outliers. Our optimization procedure, which we call **R**obust **E**ntropy **A**daptive **L**oss **M**inimization (REALM), optimizes the robust function of the entropy while simultaneously learning α (the shape of the loss), and threshold λ (the scale of the loss):

$$\theta^*,\alpha^*,\lambda^* = \min_{\theta,\alpha,\lambda} S_{\text{div}}(x)\rho\left(\mathcal{L}(\theta;x);\alpha,\lambda\right). \tag{11}$$

Further note that, under the constraint that $\alpha \in (0,2]$, the robust minimization problem Eq. (11) satisfies the SPL framework, and can be recast as the regularized problem

$$\min_{\theta}\rho\left(\mathcal{L}(\theta);\alpha,\lambda\right) = \min_{\theta,w}\left[w\frac{\mathcal{L}(\theta)}{\lambda}+g(w;\alpha)\right]. \tag{12}$$

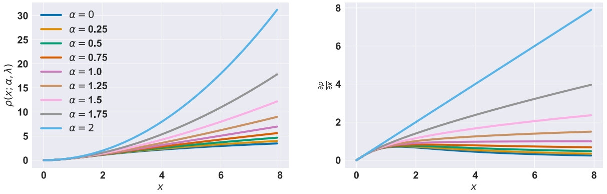


Figure 3. Robust loss (left) and its derivative (right) for varying α and fixed scale $\lambda = 1$.

The regularizer $g(w;\alpha)$ can be defined explicitly, as follows:

$$g(w;\alpha) = \frac{|\alpha-2|}{\alpha}\left[w^{\frac{\alpha}{\alpha-2}}\left(1-\frac{\alpha}{2}\right)+\frac{\alpha}{2}w-1\right]. \tag{13}$$

This formula can be obtained by considering the derivatives of Eq. (12) with respect to θ and w, and by following similar considerations as in [4]; we refer to supplementary material A for the complete derivation.

The equivalence in Eq. (12) implies that REALM performs TTA using a self-paced learning objective, and the theoretical motivation underpinning REALM is the same as that for EATA, but the regularizer chosen is less strict yielding more gradient updates during optimization over EATA.

To highlight the advantage of our proposed approach, we first note that ρ_{EATA} and $\mathcal{L}$ are both extremes on the distribution of possible scaling functions $\rho(\cdot)$. In particular, using the standard loss results in no penalization, whereas ρ_{EATA} is the most strict penalization resulting in no gradient update when the loss is high. However, visualizing $\rho(x;\alpha,\lambda)$ in Fig. 3 for the squared loss, and a range of α reveals many functions that still offer penalization of outliers without yielding zero gradient update on such outliers. $\alpha \in [0, 0.5]$ yields solutions that have a small gradient update for outlier samples, while behaving similar to the initial loss for inliers.

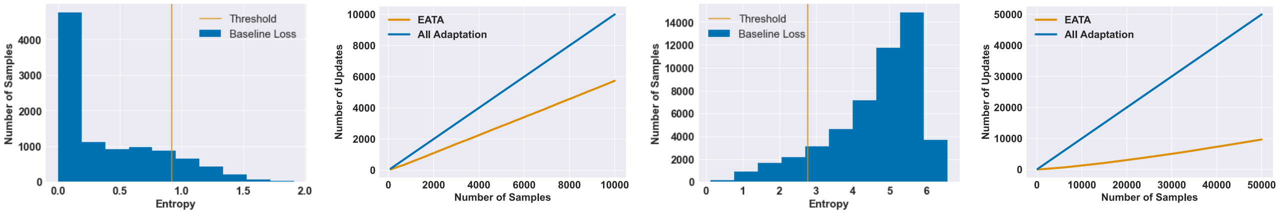
5. Experiments

We show shortcomings of TTA with entropy minimization, and demonstrate benefits of REALM over existing approaches with batch size of one. We answer the questions: (1) Does a sample reliability criterion allow for sufficient sample updates? (2) Does adapting on all samples irrespective of their reliability increase TTA performance? (3) How robust is REALM to model architecture, data quantity, and shift? Additional details are in Supplementary Material B, and ablation studies are in Supplementary Material D.

5.1. Experimental Setup

Datasets and Models We experiment with CIFAR-10-C and ImageNet-C benchmarks [15]. These datasets contain corrupted versions of the CIFAR-10 test set and ImageNet

[7] The original definition of the robust loss function appearing in [2] results in a squared loss term as the original general robust loss is intended for squared-error loss functions. Starting from that definition leads to a squared entropy objective. More details are in Supplementary Material A.



(a) C10 Loss (b) C10 EATA Updates (c) ImageNet Loss (d) ImageNet EATA Updates

Figure 4. (a), (c): Entropy values prior to adaptation for different CIFAR-10 and ImageNet. (b), (d): Number of updates over adaptation for EATA. All results are using the Gaussian noise corruption at severity 5.

validation set according to four categories of corruptions (noise, blur, weather, and digital), for a total of 15 different corruptions. We use the ResNet-26 model with group normalization following [43] for CIFAR experiments, and a ResNet-50 with group normalization following [39] for ImageNet.

Implementation Details For CIFAR experiments, we use SGD with no momentum, and a batch size of one, unless otherwise stated. For CIFAR-10 we set the learning rate to 0.005. The initial values are $\alpha = 0.15$ and $\lambda = 0.1$. For ImageNet, we use SGD with momentum of 0.9, a learning rate of 0.00025, and batch size of one. The initial α is the same, but $\lambda = 0.4 \times \log(c)$ where c is the number of classes following [39]. The learning rate is scaled to account for small batch size following [39]. We also do not update model parameters for the last block of the network following [39]. For ImageNet, the loss starts large, and setting $C = \lambda$ offsets this impact on the gradient update. For CIFAR-10, we set $C = 1$. Additional hyperparameter and algorithm details are outlined in Supplementary Material B.

5.2. Limiting Sample Updates and Adaptation Frequency

This section investigates the update frequency, and entropy of EATA, finding that EATA significantly limits the number of model updates, and has inconsistent updates across different loss distributions. For the gaussian noise corruption at severity 5 (highest severity), we first plot the entropy of all samples on both the CIFAR-10 and ImageNet corrupted datasets in Fig. 4(a) and (c). We find that the loss can be skewed in different ways, which can lead to differences in the update procedure. For the CIFAR-10 model, the entropy is naturally low, with many samples below the EATA threshold, and so are used to adapt the model. In contrast, for ImageNet the losses are high initially, leading to slow adaptation at the start of training as the loss for the majority of samples is above the threshold. It is important to note that depending on the order of the samples presented during TTA, this can lead to all samples for a particular class not being updated during adaption.

To further highlight the impact of entropy thresholding, we plot the number of updates the model makes as a function of the number of samples the model has seen using the EATA method for adaptation. A low number of samples used implies that the model is not learning from the shifted distribution, and performance will remain relatively similar to the pre-adaptation. Results are provided in Fig. 4 (b) and (d), and show that the number of updates is less than two-thirds the number of samples for CIFAR-10. This ratio is lower at the start of training where in the first 2000 samples, only about one-half of the samples are used to adapt the model. This makes the EATA weighting unreliable for a low number of adaptation steps. On ImageNet, the number of samples updated drops to around 10,000 (only 20% of the validation set).

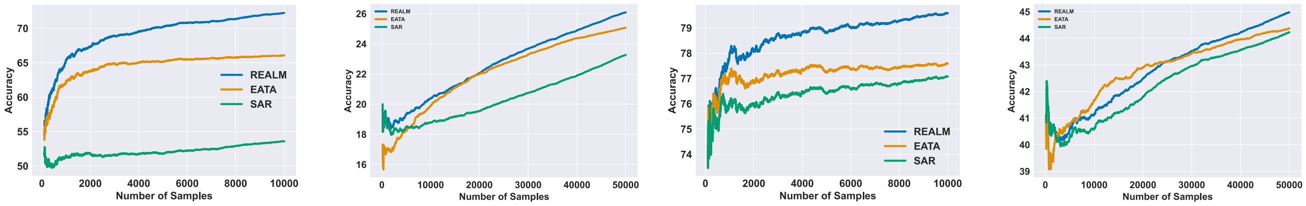
5.3. Qualitative Results for Increasing Number of Adaptation Samples

We illustrated in the previous experiments that adaptation happens relatively slowly for EATA with only $20-50\%$ of the samples being used to update the model. We now show this impacts the accuracy of online adaptation as each method progresses through test samples in CIFAR-C, and ImageNet-C in Fig. 5 for the gaussian noise and snow corruptions at severity 5.

We find that for CIFAR-10 gaussian noise, REALM achieves higher accuracy consistently at the same number of samples during adaptation by $2-4\%$ indicating faster adaptation. On ImageNet-C we note that performance is also higher at the start and end of adaptation by 1%. For the snow corruption, we note that performance increases consistently throughout adaptation for REALM. On CIFAR-10 REALM starts to outperform EATA at around 1K samples, and outperforms at around 25K samples on ImageNet reaching a net gain of $1-2\%$.

5.4. Comparison with SOTA Entropy Minimization Methods

Table 1 summarizes results comparing REALM with many SOTA methods for online and episodic TTA for a ResNet-26 with Group Normalization layers (GN) at severity 5. We find that on average, REALM performs the



(a) C10-GN (b) INet-GN (c) C10-Snow (d) INet-Snow

Figure 5. Accuracy over adaptation according to the number of samples adapted on for the gaussian noise (GN) and snow corruptions at severity 5. Accuracy is calculated as the fraction of samples correctly classified over all of the samples seen thus far. All runs use the same shuffling, and accuracy curves are averaged over 3 runs.

		Noise			Blur				Weather				Digital			
Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet26 (GN)	51.6	55.2	49.7	75.9	52.3	75.5	75.9	75.9	66.9	72.0	85.9	70.3	74.4	56.3	71.7	67.3
+ TTAug	56.6	60.4	57.1	71.7	55.3	73.7	73.7	78.6	71.5	76.7	87.9	67.1	78.3	56.8	**78.3**	69.6
+ MEMO	56.5	60.1	56.7	73.6	55.6	74.9	75.0	79.1	71.7	77.2	**88.1**	71.7	**78.9**	57.2	**78.3**	70.3
+ SFT	68.1	73.3	**71.1**	82.3	55.8	81.6	79.8	79.2	76.6	79.3	86.1	74.6	75.5	**78.1**	74.9	75.8
+ EATA	66.1	67.2	58.6	80.6	57.6	79.5	79.9	77.6	72.9	77.3	86.2	76.0	76.0	70.7	73.5	73.3
+ SAR	53.6	59.1	49.9	79.8	53.0	78.3	79.2	77.1	70.5	76.1	86.2	74.5	75.7	60.5	72.6	69.7
+ REALM	**72.2**	**74.6**	64.5	**84.5**	**62.3**	**82.6**	**83.1**	**79.6**	**77.7**	**81.0**	87.1	**82.0**	76.9	**78.0**	75.8	**77.5**

Table 1. Accuracy across all corruptions in CIFAR10-C comparing REALM with prior SOTA methods. Results for REALM are averaged over 3 runs.

best across all corruptions, outperforming EATA, SAR, and Surgical Fine-Tuning (SFT), which trains specific convolutioanl layers of a network, and uses test-time augmentations based on augmix to create an "artificial batch" following [49]. REALM consistently performs as the top method across all corruptions except impulse noise, brightness, elastic transform, and JPEG. For all corruptions, except impulse noise, REALM is still the best performing method among the online adaptation methods. For impulse noise, REALM performs in the top two, performing a bit worse than SFT. Results for MEMO, and TTAug are taken from [49], and results for SFT on gaussian noise and impulse noise corruptions are taken from [29] as we found that training with the fixed lr from our hyperparameters resulted in the model predicting the same class on all inputs for a small number of our runs[8].

Table 2 summarizes comparison of REALM with SOTA TTA methods for a ResNet-50 GN at severity 5. Results are taken from [39], however we confirmed performance for both EATA, and SAR. REALM performs the best on average across all corruptions, outperforming EATA, SAR, and DDA (which uses a diffusion model trained on the same in-distribution set). REALM is consistently a top two method across all corruptions except for frost, contrast, and jpeg compression. In these corruptions, REALM is competitive with the top performing method.

5.5. Comparisons with Different Model Architectures

We experiment with classifiers beyond the ResNet models used in the previous section to evaluate whether REALM's improvements apply to arbitrary architectures. Table 3 shows results for four different architectures on the gaussian noise corruption at severity 5 in ImageNet-C. Following [13], we select ResNet-50 GN, two transformer architectures: VitBase-LN (vision transformer with layer norm), Swin-tiny transformer, and a ConvNext-tiny network as these are all state-of-the-art attentional and convolutional architectures with roughly the same number of parameters. The Vitbase model is trained with a learning rate of $0.001/64$ following [39]. All other models are trained with the same hyperparameters as the ResNet architecture.

For both convolutional networks, REALM performs the best, while on the transformer models, REALM performs slightly worse. Both EATA and SAR perform inconsistently across the architectures with instances where they perform almost 10% worse than the best performing method. Nonetheless, we believe that no method consistently outperforms across architectures, and investigating architectural differences with TTA methods should be the subject of future work.

5.6. Comparison with Few Adaptation Samples

We experiment with adaptation of the ResNet model used in the previous section to evaluate whether REALM's improvements apply under limited sample settings. Table 4

[8] In SFT, the authors perform a large hyperparameter experiment over lr, weight decay, and steps. While our results are comparable to reported results for our choice of parameters, some small differences may cause instabilities we observed in our experiments.

	Noise			Blur				Weather				Digital				
Method	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	Avg.
ResNet50 (GN)	18.0	19.8	17.9	**19.8**	11.4	21.4	24.9	40.4	**47.3**	33.6	69.3	36.3	18.6	28.4	52.3	30.6
+ Tent	2.5	2.9	2.5	13.5	3.6	18.6	17.6	15.3	23.0	1.4	70.4	42.2	6.2	**49.2**	53.8	21.5
+ MEMO	18.5	20.5	18.4	17.1	12.6	21.8	26.9	40.4	47.0	34.4	69.5	36.5	19.2	32.1	53.3	31.2
+ DDA	**42.4**	**43.3**	**42.3**	16.6	**19.6**	21.9	26.0	35.7	40.1	13.7	61.2	25.2	**37.5**	46.6	54.1	35.1
+ EATA	24.8	28.3	25.7	18.1	17.3	28.5	29.3	44.5	44.3	41.6	70.9	**44.6**	27.0	46.8	55.7	36.5
+ SAR	23.4	26.6	23.9	18.4	15.4	28.6	**30.4**	**44.9**	44.7	25.7	**72.3**	44.5	14.8	47.0	**56.1**	34.5
+ REALM	26.9	29.9	28.0	18.4	18.2	**29.6**	**31.1**	**45.6**	43.6	**45.5**	71.2	44.4	28.9	**49.7**	55.5	**37.8**

Table 2. Accuracy across all corruptions in ImageNet-C comparing REALM with prior SOTA. Results are averaged over 3 runs.

Method	RNet-50	ViT	SWIN	ConvNext	Avg.
EATA	24.8	31.4	**38.5**	48.5	35.8
SAR	23.3	**41.0**	31.3	51.4	36.8
REALM	**26.1**	36.4	36.0	**54.8**	**38.3**

Table 3. Accuracy over four different networks: ResNet-50 with group normalization layers, ViT-base, Swin-tiny, and ConvNext-tiny pretrained on ImageNet-1k. Comparisons are done on Imagenet-C gaussian noise corruption with severity 5 and results are averaged over 3 runs.

shows results for increasing total number of samples for adaptation on the Gaussian noise corruption at severity 5 in ImageNet-C. To evaluate models adapted on a subset of the data, we holdout the last 10k samples from the test set. All models are trained with the same hyperparameters.

Method	1024	2048	4096	10k	20k
ResNet	17.7	17.7	17.7	17.7	17.7
+ EATA	**18.2**	18.3	19.0	21.4	24.6
+ SAR	17.8	17.9	18.0	19.2	21.4
+ REALM	18.1	**18.6**	**19.5**	**21.6**	**25.2**

Table 4. Results for increasing number of adaptation samples for a ResNet-50 GN. Comparisons are done on Imagenet-C gaussian noise corruption with severity 5 and results are averaged over 3 runs.

We find that REALM outperforms EATA starting from 2048 adaptation samples by around 0.5%, and outperforms SAR at all number of adaptation samples. We also note that accuracy increases consistently with increasing number of adaptation samples indicating better domain generalization capability with additional samples. Finally, we see that all methods have the largest jump in improvement at 10K and REALM reaches similar performance on the held-out set after 20k adaptation steps that EATA and SAR reach adaptation on the full validation set.

5.7. Comparisons with Different Datasets

We further conduct experiments on additional ImageNet distribution shift datasets: ImageNet-Renditions (R) [14] and ImageNet-Adversarial (A) [16]. Differing from corruption robustness in ImageNet-C, ImageNet-R and ImageNet-A contain real samples that are difficult for models trained without domain generalization properties to classify. In particular, ImageNet-R contains renditions of a subset of the classes in ImageNet including paintings, sculptures, embroidery, cartoons, origami, and toys. Imagenet-A contains images collected from iNaturalist and Flickr that are incorrectly classified by a ResNet-50. Additional details about the dataset are available in the supplementary material.

Method	ImageNet-R	ImageNet-A
No Adapt	40.8	0.1
REALM	**42.5**	**14.3**

Table 5. Results for ResNet-50 GN evaluated on the ImageNet-R and ImageNet-A datasets.

Results on these datasets for REALM are shown in Tab. 5 indicating REALM improves performance on both datasets over no adaptation, and that REALM improves performance on distribution shifts outside common corruptions.

6. Conclusion

This work illustrates the shortcomings of TTA in the online single instance batch setting. We highlight that current approaches aimed at stabilizing TTA by skipping unreliable samples with high entropy, result in no adaptation on a large portion of the dataset. We then show equivalence to SPL with a specific regularizer, and introduce REALM our approach for stabilizing online TTA entropy minimization approaches. REALM improves on prior approaches by penalizing the update of all samples using a robust function of the entropy rather than skipping the sample based on entropy entirely. This yields improved results on corruptions of CIFAR-10 and ImageNet. Additionally, REALM is simple to implement, requiring only modification of the loss function, and is theoretically grounded within the framework of self-paced learning. We believe this work is a step towards creating more robust models through online TTA.

Acknowledgements

We are grateful to Zakaria Aldeneh, Masha Fedzechkina, and Russ Webb for their helpful discussions, comments, and thoughtful feedback in reviewing this work.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[3] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 494–503, 2021.

[4] Michael Black and Anand Rangarajan. On the unification line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19:57–91, 07 1996.

[5] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[6] Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8344–8353, 2022.

[7] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.

[8] John E Dennis Jr and Roy E Welsch. Techniques for nonlinear least squares and robust regression. *Communications in Statistics-simulation and Computation*, 7(4):345–359, 1978.

[9] Antonio D'Innocente, Francesco Cappio Borlino, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. One-shot unsupervised cross-domain detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 732–748. Springer, 2020.

[10] Yanbo Fan, Ran He, Jian Liang, and Baogang Hu. Self-paced learning: An implicit regularization perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[11] François Fleuret et al. Test time adaptation through perturbation robustness. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

[13] Jin Gao, Jialing Zhang, Xihui Liu, Trevor Darrell, Evan Shelhamer, and Dequan Wang. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11786–11796, 2023.

[14] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[15] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[16] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15262–15271, June 2021.

[17] Melvin J Hinich and Prem P Talwar. A simple method for robust regression. *Journal of the American Statistical Association*, 70(349):113–119, 1975.

[18] Peter J Huber. Robust estimation of a location parameter. *Breakthroughs in statistics: Methodology and distribution*, pages 492–518, 1992.

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[20] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[21] Mengmeng Jing, Xiantong Zhen, Jingjing Li, and Cees Snoek. Variational model perturbation for source-free domain adaptation. *Advances in Neural Information Processing Systems*, 35:17173–17187, 2022.

[22] Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021.

[23] Changhun Kim, Joonhyung Park, Hajin Shim, and Eunho Yang. Sgem: Test-time adaptation for automatic speech recognition via sequential-level generalized entropy minimization. *arXiv preprint arXiv:2306.01981*, 2023.

[24] Hiroaki Kingetsu, Kenichi Kobayashi, Yoshihiro Okawa, Yasuto Yokota, and Katsuhito Nakazawa. Multi-step test-time adaptation with entropy minimization and pseudo-labeling. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 4153–4157. IEEE, 2022.

[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[26] Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

[27] M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23, 2010.

[28] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4544–4553, 2020.

[29] Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

[30] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9641–9650, 2020.

[31] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016.

[32] Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling SGD with stochastic differential equations (sdes). In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12712–12725, 2021.

[33] Jian Liang, Ran He, and Tieniu Tan. A comprehensive survey on test-time adaptation under distribution shifts. *arXiv preprint arXiv:2303.15361*, 2023.

[34] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

[35] Guan-Ting Lin, Shang-Wen Li, and Hung-yi Lee. Listen, adapt, better wer: Source-free single-utterance test-time adaptation for automatic speech recognition. *arXiv preprint arXiv:2203.14222*, 2022.

[36] Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 34:21808–21820, 2021.

[37] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020.

[38] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022.

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

[40] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33:11539–11551, 2020.

[41] Vatsal Shah, Xiaoxia Wu, and Sujay Sanghavi. Choosing the sample with lowest loss makes sgd robust. In *International Conference on Artificial Intelligence and Statistics*, pages 2120–2130. PMLR, 2020.

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

[43] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020.

[44] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.

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

[46] Steven Wegmann, Puming Zhan, Ira Carp, Michael Newman, Jon Yamron, and Larry Gillick. Dragon systems' 1998 broadcast news transcription system. In *EUROSPEECH*. Citeseer, 1999.

[47] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[48] Puning Yang, Jian Liang, Jie Cao, and Ran He. Auto: Adaptive outlier optimization for online test-time ood detection. *arXiv preprint arXiv:2303.12267*, 2023.

[49] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022.

[50] Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. *arXiv preprint arXiv:2306.03536*, 2023.

[51] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.