

# On the Robustness of Open-World Test-Time Training: Self-Training with Dynamic Prototype Expansion

Yushu Li<sup>1</sup> Xun Xu<sup>2†</sup> Yongyi Su<sup>1</sup> Kui Jia<sup>1†</sup>

<sup>1</sup> South China University of Technology

<sup>2</sup> Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A\*STAR)

{eeyushuli, eesuyongyi}@mail.scut.edu.cn, {alex.xun.xu, kuijia}@gmail.com

## Abstract

Generalizing deep learning models to unknown target domain distribution with low latency has motivated research into test-time training/adaptation (TTT/TTA). Existing approaches often focus on improving test-time training performance under well-curated target domain data. As figured out in this work, many state-of-the-art methods fail to maintain the performance when the target domain is contaminated with strong out-of-distribution (OOD) data, a.k.a. open-world test-time training (OWTTT). The failure is mainly due to the inability to distinguish strong OOD samples from regular weak OOD samples. To improve the robustness of OWTTT we first develop an adaptive strong OOD pruning which improves the efficacy of the self-training TTT method. We further propose a way to dynamically expand the prototypes to represent strong OOD samples for an improved weak/strong OOD data separation. Finally, we regularize self-training with distribution alignment and the combination yields the state-of-the-art performance on 5 OWTTT benchmarks. The code is available at <https://github.com/Yushu-Li/OWTTT>.

## 1. Introduction

The distribution gap between training and testing data poses great challenges to the generalization of modern deep learning methods [34, 4]. To improve model’s generalization to testing data which may feature a different data distribution from the training data, domain adaptation has been extensively studied [44] to learn domain invariant features. Nevertheless, the existing unsupervised domain adaptation paradigm requires simultaneous access to both source and target domain data with an off-line training stage [12, 41]. In a realistic scenario, access to target domain data may not become available until the inference stage, and instant

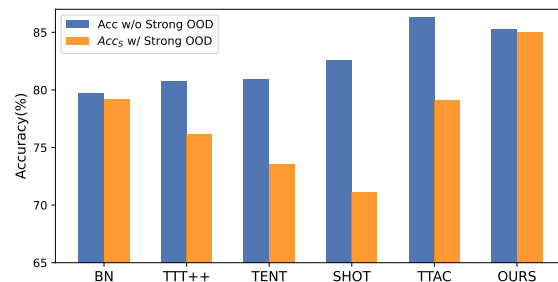


Figure 1: Existing test-time training methods suffer substantially when target domain data is contaminated with strong OOD samples. We illustrate the results by test-time training on CIFAR10-C contaminated by SVHN as strong OOD samples, with reference to values extracted from Tab 1.

prediction on testing data is required without further ado. Therefore, these requirements give rise to the emergence of a new paradigm of adaptation at test time, a.k.a. test-time training/adaptation (TTT/TTA) [40, 43].

The success of TTT has been demonstrated on many synthesized corrupted target domain data [18], manually selected hard samples [35] and adversarial samples [8]. Nevertheless, the boundary of existing TTT methods’ capability is yet to be fully explored. In particular, to enable TTT in the open-world, focus has been shifted to investigating open-world scenarios where TTT methods could fail. Among these scenarios, when testing data is not drawn in an i.i.d. fashion, TTT methods may be biased towards the continually changing distribution [45, 14]. When the target domain consists of testing data drawn from both source and target distributions, [29] developed a non-forgetting training paradigm to avoid failure on source domain samples. When TTT must be updated with small batch size, shifting distribution, and class imbalanced testing data, [30] proposed to swap out the batch normalization and remove unreliable pseudo labels for improved robustness.

Despite many efforts into developing stable and robust TTT methods under a more realistic open-world environ-

<sup>†</sup>Correspondence to Xun Xu: alex.xun.xu@gmail.com and Kui Jia: kuijia@gmail.com.

ment, in this work, we delve into an overlooked, but very commonly seen open-world scenario where the target domain may contain testing data drawn from a significantly different distribution, e.g. different semantic classes than source domain, or simply random noise. We refer to the above testing data as strong out-of-distribution (strong OOD) data, as opposed to distribution-shifted testing data, e.g. common corruptions, which are referred to as weak OOD data in this work. The ignorance of this realistic setting by existing works, thus, drives us to explore improving the robustness of open-world test-time training (OWTTT) where testing data is contaminated with strong OOD samples.

As shown in Fig. 1, we first empirically evaluate existing TTT methods and reveal that TTT methods through self-training [43, 25] and distribution alignment [27, 39] would all suffer substantially subject to strong OOD samples. These results suggest applying existing TTT techniques fails to achieve safe test-time training in the open-world. We attribute the failure to the following two reasons. First, self-training based TTT [25] struggles to deal with strong OOD samples as it has to assign testing samples to known classes. Despite some low confident samples that could be filtered out by applying a threshold as adopted in semi-supervised learning [38], it is still not guaranteed to prune out all strong OOD samples. This issue exacerbates when strong OOD samples are not “strong enough”. For example, when adapting the model pre-trained on CIFAR10 to CIFAR10-C, testing samples drawn from CIFAR100-C are significantly more difficult to be pruned out than random noise samples. Thereby, learning from incorrectly labeled strong OOD samples would weaken the model’s discrimination on weak OOD samples. Second, distribution alignment based approaches would suffer when strong OOD samples are counted for estimating target domain distribution. Both global distribution alignment [27] and category-wise distribution alignment [39] could be affected and lead to less accurate feature distribution alignment.

With the potential causes for the failure of existing TTT methods in mind, we propose two techniques to improve the robustness of open-world TTT under a self-training framework. First, we build the baseline of TTT upon a variant of self-training, i.e. clustering in the target domain with source domain prototypes as cluster centers. To lessen the influence of self-training with incorrectly pseudo-labeled strong OOD samples, we design a hyper-parameter-free method to prune out strong OOD samples. To further separate the features of weak and strong OOD samples, we allow the prototype pool to expand by including isolated strong OOD samples. Therefore, self-training will allow strong OOD samples to form tight clusters around newly expanded strong OOD prototypes. Second, as strong OOD samples are likely to be attracted by newly grown prototypes, a more accurate

distribution characterizing the target domain data can be estimated. This would benefit distribution alignment between source and target domains. Therefore, we further propose to regularize self-training with global distribution alignment to reduce the risk of confirmation bias [3]. Finally, to synthesize an open-world TTT scenario, we employ CIFAR10-C, CIFAR100-C, ImageNet-C, VisDA-C, ImageNet-R, Tiny-ImageNet, MNIST, and SVHN datasets and create a benchmark by treating one dataset as weak OOD and others as strong OOD. We refer to this benchmark as the open-world test-time training benchmark and hope this would encourage more future works to pay attention to the robustness of test-time training in more realistic scenarios.

We summarize the contributions of this work below.

- Overlooked by existing studies into test-time training, we argue that open-world test-time training (OWTTT) could be spoiled by strong OOD testing data. We demonstrated that without special treatment state-of-the-art TTT methods fail to generalize well under the open-world protocol.
- We introduce a baseline method by prototype clustering with distribution alignment regularization. A strong OOD detector and prototype expansion are further developed to improve the robustness of the baseline under OWTTT protocol.
- We established a benchmark for evaluating OWTTT protocol covering multiple types of domain shift, including common corruptions and style transfer. Our approach achieves state-of-the-art performance on the proposed benchmark.

## 2. Related Work

### 2.1. Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) [12] aims to improve models’ ability to generalize to target domain data where no labeled data exists. UDA is often achieved by learning invariant features across source and target domains [12], discovering cluster structures in the target domain [41, 46, 7], self-supervised training [26], distance-based alignment [22], etc. Although UDA has made considerable progress in improving the generalizability of the target domain, having access to both the source and target domain during adaptation is not always realistic, e.g. due to data privacy issues. Source-free domain adaptation (SFDA) [25, 33, 10] has thus emerged which gets rid of the access to source domain data. Nevertheless, SFDA is still not versatile enough to deal with a more realistic scenario where target domain distribution is not known before testing begins.

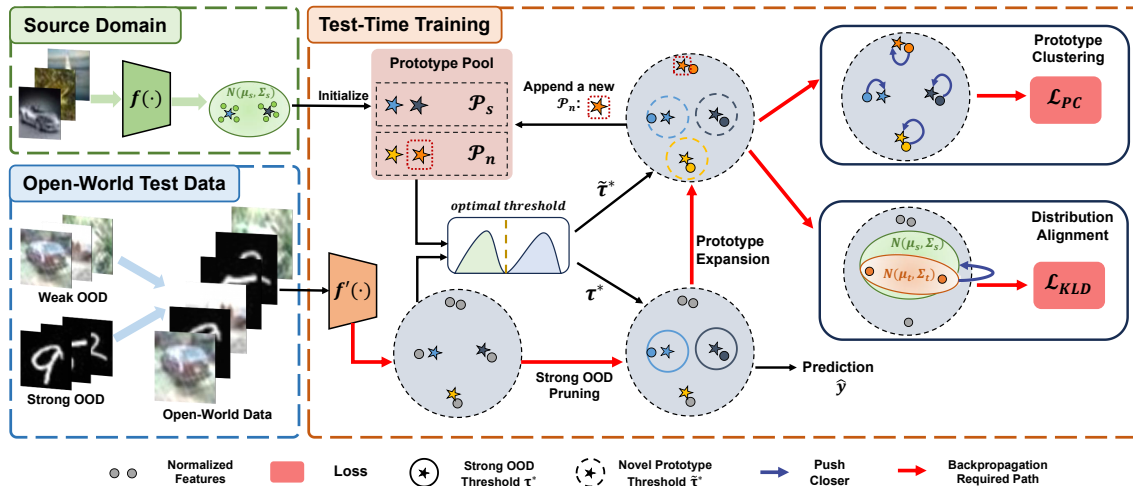


Figure 2: An overview of the proposed method for open-world test-time training. Testing data may be contaminated with strong OOD samples. We first prune out potential strong OOD samples by  $\tau^*$  for inference. Then, we expand the prototype pool by isolating strong OOD samples and update model weights through self-training and distribution alignment.

## 2.2. Test-Time training

Considering that in some scenarios we would like models that have been deployed to the target domain to automatically adapt to the new environment without accessing source domain data. With these considerations in mind, In response to the demand for adaptation to arbitrary unknown target domain with low inference latency, test-time training/adaptation (TTT/TTA) [40, 43] have emerged. TTT is often realized by three types of paradigms. Self-supervised learning on the testing data enables adapting to the target domain without considering any semantic information [40, 27]. Self-training reinforces model’s prediction on unlabeled data and has been demonstrated to be effective for TTT [43, 6, 25, 15]. Lastly, distribution alignment provides another viable approach towards TTT by adjusting model weights to produce features following the same distribution as the source domain [39, 27]. Despite the efforts into developing more sophisticated TTT methods, certifying the robustness of TTT is still yet to be fully investigated. Recent works studied the robustness of TTT when target domain distribution shifts over time [45, 14]. A more extensive investigation into TTT under small batch size and imbalanced classes was carried out [30]. Orthogonal to the existing attempts into robustifying TTT, we revealed that TTT is extremely vulnerable to an open-world scenario where testing data consists of strong OOD samples. In this work, we aim to improve the robustness of open-world test-time training by self-training with a dynamic expanded prototype pool.

## 2.3. Open-Set Domain Adaptation

[37] introduced the concept of open-set recognition, referring to a setting in that trained models are required to

reject testing samples drawn from unknown semantic categories. In the context of domain adaptation, ATI [5] proposed open-set domain adaptation (OSDA) and implemented open-set identification by defining and maximizing the open-set to closed-set distance. OSBP [36] uses backpropagation methods to make the logits of the unknown class samples into a recognizable constant by joint adversarial training of the generator and classifier. DAOD [11] offers a new perspective by proposing unsupervised open-set domain adaptation (UOSDA) where the target domain has unknown classes that are not found in the source domain. [47] moved towards a soft rejection method for open-set domain adaptation. In the open-world setting we advocate, we assume testing data could be contaminated by arbitrary strong OOD data, thus it makes little sense to discover new categories for strong OOD data. Instead, we define open-world test-time training as classifying testing samples from source domain categories and rejecting any unknown OOD sample.

## 3. Methodology

In this section, we first overview the formulation of test-time training. Then, we introduce TTT by prototype clustering and how to expand prototypes for open-world test-time training. Distribution alignment is finally combined with prototype clustering to achieve robust open-world test-time training. An overview of the method is presented in Fig. 2.

### 3.1. Overview of Test-Time Training

We first briefly review the practice of test-time training/adaptation for classification tasks. Test-time training aims to adapt the source domain pre-trained model to the target domain which may be subject to a distribution shift

from the source domain. We first give an overview of the self-training based TTT paradigm, following the protocol defined in [39]. In specific, we define the source and target domain datasets as  $\mathcal{D}_s = \{x_i, y_i\}_{i=1 \dots N_s}$  with label space  $\mathcal{C}_s = \{1 \dots K_s\}$  and  $\mathcal{D}_t = \{x_i, y_i\}_{i=1 \dots N_t}$  with label space  $\mathcal{C}_t = \{1 \dots K_s, K_s + 1, \dots, K_s + K_t\}$ . In closed-world TTT, the two label spaces are identical while  $\mathcal{C}_s \subseteq \mathcal{C}_t$  is true under open-world TTT. At the testing stage, a mini-batch of testing samples at timestamp  $t$  is denoted as  $\mathcal{B}_t$ . We further denote the representation learning network as  $z_i = f(x_i; \Theta) \in \mathbb{R}^D$  and the classifier head as  $h(z_i; \omega, \beta)$ . Test-time training is achieved by updating the representation network and/or classifier parameters on the target domain dataset  $\mathcal{D}_t$ . To avoid confusion between TTT definitions, we adopt the sequential test-time training (sTTT) proposed in [39] for evaluation. Under the sTTT protocol, testing samples are sequentially tested and a model update is carried out after a minibatch of testing samples is observed. The prediction on any testing sample arriving at time-stamp  $t$  will not be affected by any testing samples arriving at  $t+k$  where  $k \geq 1$ .

### 3.2. TTT by Prototype Clustering

Inspired by the success of discovering clusters in domain adaptation tasks [41, 36], we formulate test-time training as discovering cluster structures in the target domain data. The cluster structures are identified in the target domain by identifying representative prototypes as cluster centers and testing samples are encouraged to embed close to one of the prototypes. Inference is then enabled by measuring the testing samples' similarity to the prototypes in the feature space. Formally, we write the prototypes in the source domain as  $\mathcal{P}_s = \{p_k \in \mathbb{R}^D\}_{k \in \mathcal{C}_s}$ . The prototype clustering objective is defined as minimizing the following negative log-likelihood loss, where  $\hat{y}$  indicates the pseudo label  $\hat{y} = \arg \max_k h(f(x))$  for  $z_i$  and  $\langle \cdot, \cdot \rangle$  measures the cosine similarity.

$$\mathcal{L}_{PC} = - \sum_{k \in \mathcal{C}_s} \mathbb{1}(\hat{y} = k) \log \frac{\exp(\frac{\langle p_k, z_i \rangle}{\delta})}{\sum_l \exp(\frac{\langle p_l, z_i \rangle}{\delta})} \quad (1)$$

Minimizing the above objective will allow testing samples to embed close to their predicted prototypes and away from other prototypes. In a closed-world test-time training scenario [39], prototype clustering has demonstrated strong performance [25]. Nevertheless, prototype clustering is severely challenged in the open-world test-time training scenario where strong OOD samples may exist. Should strong OOD samples be forcibly categorized into any source category, self-training on the noisy labeled sample would confuse the network's discriminative capability on weak OOD samples.

**Strong OOD Pruning:** We develop a hyper-parameter-free approach to prune out strong OOD samples to avoid the

negative impact of adapting model weights. Specifically, we define a strong OOD score  $os_i$  for each testing sample as the highest similarity to source domain prototypes as in Eq. 2.

$$os_i = 1 - \max_{p_k \in \mathcal{P}_s} \langle f(x_i), p_k \rangle \quad (2)$$

We make an observation that the outlier score is subject to a bimodal distribution as shown in Fig. 3. Therefore, instead of specifying a fixed threshold, we define the optimal threshold as separating the two distribution modalities. In specific, the problem can be formulated as dividing the outlier scores into two clusters and the optimal threshold will minimize the intra-cluster variations in Eq 3, where  $N^+ = \sum_i \mathbb{1}(os_i > \tau)$  and  $N^- = \sum_i \mathbb{1}(os_i \leq \tau)$ . Optimizing Eq 3 can be efficiently implemented by exhaustively searching over all possible thresholds from 0 to 1 with a step of 0.01. To maintain a stable estimation of the outlier score distribution, we update the distribution in an exponential moving average manner, with length  $N_m$ .

$$\min_{\tau} \frac{1}{N^+} \sum_i [os_i - \frac{1}{N^+} \sum_j \mathbb{1}(os_j > \tau) os_j]^2 + \frac{1}{N^-} \sum_i [os_i - \frac{1}{N^-} \sum_j \mathbb{1}(os_j \leq \tau) os_j]^2 \quad (3)$$

With the optimal threshold  $\tau^*$ , we could identify strong OOD samples with two benefits. First, during the model weights update, we can exclude detected OOD samples from self-training w.r.t. source domain prototypes. Second, it provides us with a way to differentiate weak OOD samples from strong ones during the inference stage.

### 3.3. Open-World TTT by Prototype Expansion

Identifying strong OOD samples and excluding them from updating model weights does not guarantee a good separation between weak OOD testing samples from strong OOD ones. Inspired by the success of novelty detection [32], we propose to dynamically expand the prototype pool to incorporate prototypes representing strong OOD samples. Self-training is then applied with both source domain prototypes and strong OOD prototypes to create wider gaps between weak and strong OOD samples in the feature space. Specifically, we denote additionally a strong OOD prototype set as  $\mathcal{P}_n = \{p_k \in \mathbb{R}^D\}_{k \in \mathcal{C}_t \setminus \mathcal{C}_s}$ . When no prior information on the target domain is available, we initialize the novel prototypes as an empty set. As TTT goes on,  $\mathcal{P}_n$  is expected to expand to accommodate the unknown distribution in the target domain.

**Prototype Expansion:** Expanding a strong OOD prototype pool requires evaluating testing samples against both source domain and strong OOD prototypes. A similar problem was investigated for the purpose of dynamically esti-

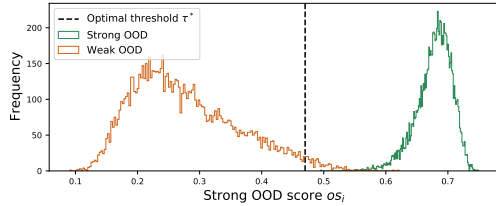


Figure 3: A bimodal distribution of strong OOD scores is observed. An optimal threshold separates weak and strong OOD samples.

imating the number of clusters from data. A deterministic hard clustering algorithm, DP-means [21], was developed by measuring the distance of data points to known cluster centers, and a new cluster will be initialized when the distance is above a threshold. DP-means is shown to be equivalent to optimizing a K-means objective with an additional penalty on the number of clusters. DP-means was later adapted to few-shot learning tasks [2]. DP-means provides a viable solution to dynamic prototype expansion with an additional threshold hyper-parameter  $\tau$ , which can be estimated from the concentration parameter  $\alpha$  for the Chinese restaurant process [1]. Despite the connection to theoretical stochastic processing, estimating  $\tau$  requires knowing the standard deviation for the base distribution which is still not trivial. To ease the difficulty of estimating additional hyper-parameter, we first define a testing sample with an extended strong OOD score  $\tilde{o}s_i$  as the closest distance to existing source domain prototypes and strong OOD prototypes as in Eq. 4. A dynamic threshold is estimated following Eq. 3. Hence, testing samples above this threshold will establish a new prototype. To avoid adding close-by testing samples, we repeat the additive process incrementally. Since model weights are subject to constant updates, OOD prototypes may experience a representation shift. To avoid expanding OOD prototypes indefinitely, we treat the  $\mathcal{P}_n$  as a queue with fixed length  $N_q$ , and old OOD prototypes will be discarded when new OOD prototypes are appended.

$$\tilde{o}s_i = 1 - \max_{p_k \in \mathcal{P}_s \cup \mathcal{P}_n} \langle f(x_i), p_k \rangle \quad (4)$$

**Prototype Clustering with Strong OOD Prototypes:** With additional strong OOD prototypes being identified, we define the prototype clustering loss for testing samples with two considerations. First, testing samples classified as known classes should embed closer to the prototype and keep away from other prototypes, which defines a K-way classification task. Second, testing samples classified as strong OOD prototypes should embed further away from any source domain prototypes, which defines a K+1-way classification task. With these objectives in mind, we define the prototype clustering loss as in Eq. 5, where the pseudo labels are predicted as  $\hat{y}_i = \arg \max_{p_k \in \mathcal{P}_s \cup \mathcal{P}_n} \langle p_k, z_i \rangle$ .

$$\begin{aligned} \mathcal{L}_{PC} = & - \sum_{k \in \mathcal{C}_s} \mathbb{1}(\hat{y}_i = k) \log \frac{\exp(\frac{\langle p_k, z_i \rangle}{\delta})}{\sum_{l \in \mathcal{C}_s} \exp(\frac{\langle p_l, z_i \rangle}{\delta})} \\ & - \sum_{k \in \mathcal{C}_t} \mathbb{1}(\hat{y}_i = k) \log \frac{\exp(\frac{\langle p_k, z_i \rangle}{\delta})}{\sum_{l \in \mathcal{C}_s+1} \exp(\frac{\langle p_l, z_i \rangle}{\delta})} \end{aligned} \quad (5)$$

### 3.4. Distribution Alignment Regularization

Self-training (ST) is known to be prone to incorrect pseudo labels, a.k.a. confirmation bias [3]. The situation exacerbates when the target domain consists of OOD samples. To reduce the risk of ST failure, we further incorporate distribution alignment [27] as regularization to self-training. Specifically, we assume a Gaussian distribution  $\mathcal{N}(\mu_s, \Sigma_s)$  for the source domain feature. In the target domain, we incrementally estimate the feature distribution  $\mathcal{N}(\mu_t, \Sigma_t)$  with a momentum  $\beta$  [39]. Only testing samples passing the strong OOD pruning are included for estimating the distribution. The KL-Divergence loss  $\mathcal{L}_{KLD}$  between source and target domain distributions is finally utilized to regularize prototype clustering on the target domain.

$$\mathcal{L}_{KLD} = D_{KL}(\mathcal{N}(\mu_s, \Sigma_s) || \mathcal{N}(\mu_t, \Sigma_t)) \quad (6)$$

### 3.5. Open-World TTT Algorithm

In this section, we provide an overview of the open-world TTT algorithm in Algo. 1. We divide the TTT procedure into two steps, the inference stage and the adaption stage.

**Inference Stage:** i) We evaluate the testing samples' strong OOD score  $os_i$  by Eq. 2 and categorize the samples into weak or strong OOD. ii) We further categorize weak OOD samples into one of the  $\mathcal{C}_s$  classes, strong OOD samples are simply counted as one separate strong OOD class.

**Adaptation Stage:** i) We calculate the extended strong OOD score and threshold and expand the prototype pool by the isolated strong OOD sample. ii) We calculate test-time training loss  $\mathcal{L}_{PC}$  and  $\mathcal{L}_{KLD}$  and update model by gradient descent. For test-time training on the full testing data, we repeat the two stages until the testing data is exhausted.

## 4. Experiments

We validate the effectiveness of open-world TTT and extensively benchmarked multiple state-of-the-art TTT methods. In the following of this section, we first elaborate on the datasets, evaluation metrics, and competing methods. Then, we present open-world TTT results on five datasets and analyze the effectiveness of individual components.

### 4.1. Datasets

We introduce several datasets, including corruption datasets, style transfer datasets, and some other com-

---

**Algorithm 1: Open-World TTT Algorithm**

---

**input** : Testing sample batch  $\mathcal{B}^t = \{x_i\}_{i=1 \dots N_B}$ 

# Inference Stage:

**for**  $x_i \leftarrow 1$  **to**  $N_B$  **do**    Calculate  $os_i$  &  $\tau^*$  by Eq. 2 & 3;

# Label Prediction:

**if**  $os_i < \tau^*$  **then**        |  $\hat{y}_i = \arg \max_{p_k \in \mathcal{P}_s} < f(x_i; \Theta), p_k >$     **else**        |  $\hat{y}_i = |\mathcal{C}_s| + 1$ 

# Adaptation Stage:

**for**  $x_i \leftarrow 1$  **to**  $N_B$  **do**    Calculate  $\tilde{os}_i$  &  $\tilde{\tau}^*$  by Eq. 4 & 3;

# Prototype Expansion:

**if**  $\tilde{os}_i > \tilde{\tau}^*$  **then**        |  $\mathcal{P}_n = \mathcal{P}_n \cup f(x_i; \Theta)$ **for**  $x_i \leftarrow 1$  **to**  $N_B$  **do**    Calculate losses  $\mathcal{L}_{PC}$  &  $\mathcal{L}_{KLD}$  by Eq. 5 & 6;    Update  $\Theta = \Theta - \alpha(\nabla \mathcal{L}_{PC} + \lambda \nabla \mathcal{L}_{KLD})$ ;

mon datasets. For the corruption datasets, We selected **CIFAR10-C/CIFAR100-C** [18] as a small corruption dataset, each containing 10,000 corrupt images with 10/100 categories, and **ImageNet-C** [18] as a large-scale corruption dataset, which contains 50,000 corruption images within 1000 categories. We also introduced some style transfer datasets. **VisDA-C** [31] is a synthetic to real large-scale dataset containing 152,397 synthetic training images and 55,388 real testing images, belonging to 12 categories. **ImageNet-R** [17] is a large-scale realistic style transfer dataset that has renditions of 200 ImageNet classes resulting in 30,000 images. **Tiny-ImageNet** [23] consists of 200 categories with each category containing 500 training images and 50 validation images. We also introduce some digits datasets. **MNIST** [24] is a handwritten digit dataset, which contains 60,000 training images and 10,000 testing images. **SVHN** [28] is a digital dataset in a real street context, including 50,000 training images and 10,000 testing images.

## 4.2. Evaluation Metric

To evaluate open-world test-time training, we adopt an evaluation metric similar to open-set classification tasks [13]. The major concern for TTT evaluation is the accuracy of source domain categories and whether strong OOD samples can be rejected. Therefore, we define two accuracy metrics, the first measures the accuracy of source domain classes, denoted  $Acc_S$ , which is equivalent to existing TTT metrics. We further define the accuracy on strong

OOD categories, denoted as  $Acc_N$ , as a binary classification task, i.e. true positive is defined as successfully rejecting a strong OOD sample as any of the source domain classes. As we pursue a good trade-off between source accuracy and OOD accuracy, we further report a harmonic mean  $Acc_H$  between  $Acc_S$  and  $Acc_N$  to measure the balanced prediction. More detailed definitions are given in Eq. 7, where  $\hat{y}_i$  refers to the predicted label and  $\mathbb{1}(y_i \in \mathcal{C}_s)$  is true if  $y_i$  is in the set  $\mathcal{C}_s$ .

$$Acc_S = \frac{\sum_{x_i, y_i \in \mathcal{D}_t} \mathbb{1}(y_i = \hat{y}_i) \cdot \mathbb{1}(y_i \in \mathcal{C}_s)}{\sum_{x_i, y_i \in \mathcal{D}_t} \mathbb{1}(y_i \in \mathcal{C}_s)}$$
$$Acc_N = \frac{\sum_{x_i, y_i \in \mathcal{D}_t} \mathbb{1}(\hat{y}_i \in \mathcal{C}_t \setminus \mathcal{C}_s) \cdot \mathbb{1}(y_i \in \mathcal{C}_t \setminus \mathcal{C}_s)}{\sum_{x_i, y_i \in \mathcal{D}_t} \mathbb{1}(y_i \in \mathcal{C}_t \setminus \mathcal{C}_s)} \quad (7)$$
$$Acc_H = 2 \cdot \frac{Acc_S \cdot Acc_N}{Acc_S + Acc_N}$$

## 4.3. Open-World Test-Time Training Protocol

We propose the Open-World Test Time Training (OWTTT) protocol to evaluate different TTT methods' performance in the open-world case. In order to set up a fair comparison with existing methods, we take all the classes in the TTT benchmark dataset as seen classes and add additional classes from additional datasets as unseen classes. By doing so we do not need to modify the source domain training process. In the later experiments, we set the number of known class samples and the number of unknown class samples to be the same. Then we follow the "One Pass" protocol in [39], which has two main restrictions. Firstly, the training objective cannot be changed during the source domain training procedure (e.g. add additional self-training branches). Secondly, testing data in the target domain is sequentially streamed and predicted.

## 4.4. Training Details

We followed the sequential test-time training protocol specified in [39] and choose ResNet-50 [16] as the backbone network for all experiments. For optimization, we choose SGD with momentum to optimize the backbone network. We set learning rate  $\alpha = \{1e-3, 1e-4, 2.5e-5, 2.5e-5, 2.5e-5\}$ , batch size  $N_B = \{256, 256, 128, 128, 128\}$ ,  $\lambda = \{1, 1, 0.4, 0.4, 0.04\}$ , respectively for experiments on Cifar10-C, Cifar100-C, ImageNet-C, ImageNet-R and VisDA-C, respectively. To further reduce the effect of incorrect pseudo labeling, we only use 50% of samples with  $od_i$  far from  $\tau^*$  to perform prototype clustering for each batch. For all experiments, we use temperature scaling  $\delta = 0.1$ , the length of strong OOD prototypes queue  $N_q = 100$ , and the length of moving average  $N_m = 512$ .

Table 1: Open-world test time training results on CIFAR10-C. All numbers are in %.

Method	Noise			MNIST			SVHN			Tiny-ImageNet			CIFAR100-C		
	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
TEST	68.59	<b>99.97</b>	81.36	60.48	88.81	71.96	60.94	86.44	71.48	57.41	79.63	66.72	52.74	74.24	61.67
BN	76.63	95.69	85.11	76.15	95.75	84.83	79.18	<b>94.71</b>	<b>86.25</b>	67.66	<b>82.67</b>	74.42	68.44	81.38	74.35
TTT++	41.09	57.31	47.86	59.52	77.52	67.34	68.77	85.80	76.34	66.70	79.28	72.44	65.69	77.47	71.10
TENT	32.24	33.30	32.77	55.64	68.27	61.31	66.70	82.50	73.77	66.54	79.32	72.37	64.80	76.40	70.12
SHOT	63.54	71.37	67.23	56.92	53.26	55.03	70.01	72.58	71.27	67.78	82.25	<b>74.32</b>	67.73	72.87	70.21
TTAC	64.46	77.42	70.35	<u>77.60</u>	84.53	80.92	77.30	81.10	79.16	<u>71.64</u>	77.14	74.29	71.94	75.44	73.65
OURS	<b>85.46</b>	<u>98.60</u>	<b>91.56</b>	<b>83.89</b>	<b>97.83</b>	<b>90.32</b>	<b>84.99</b>	<u>87.94</u>	<b>86.44</b>	<b>71.77</b>	<b>84.71</b>	<b>77.70</b>	<b>74.08</b>	<b>84.64</b>	<b>79.01</b>

Table 2: Test time training results on CIFAR100-C.

Method	Noise			MNIST			SVHN			Tiny-ImageNet			CIFAR10-C		
	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
TEST	36.75	<b>99.87</b>	53.73	25.99	49.59	34.11	30.01	81.62	43.89	25.41	70.06	37.3	25.55	73.28	37.89
BN	50.21	<u>98.72</u>	66.56	36.21	84.69	50.73	45.69	<u>90.45</u>	60.71	34.88	<b>82.18</b>	48.97	37.00	83.54	51.28
TTT++	23.47	70.26	35.19	28.31	<b>86.74</b>	42.68	37.56	90.45	53.08	34.67	81.25	48.60	33.78	81.12	47.70
TENT	22.57	66.60	33.72	27.85	80.92	41.43	37.08	89.90	52.51	35.51	77.34	48.68	35.20	80.26	48.94
SHOT	<u>51.52</u>	98.21	<u>67.58</u>	35.35	81.71	49.35	45.87	89.72	60.70	35.72	81.11	49.59	38.00	82.13	51.96
TTAC	51.11	98.66	67.34	<u>37.78</u>	<u>86.66</u>	<u>52.62</u>	<u>47.29</u>	<b>91.42</b>	<u>62.33</u>	32.04	80.46	45.83	38.83	<b>83.68</b>	53.05
OURS	<b>56.76</b>	97.25	<b>71.68</b>	<u>40.77</u>	82.91	<b>54.66</b>	<b>54.32</b>	81.98	<b>65.34</b>	<b>38.90</b>	<u>81.92</u>	<b>52.75</b>	<b>38.97</b>	83.20	<b>53.08</b>

Table 3: Test time training results on ImageNet-C.

Method	noise			MNIST			SVHN		
	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
TEST	18.51	<b>100.00</b>	31.24	18.66	<b>98.27</b>	31.36	18.94	<u>87.75</u>	31.15
BN	36.34	99.97	53.31	<u>30.77</u>	74.53	<u>43.55</u>	33.26	84.54	<u>47.74</u>
TENT	22.54	10.47	14.29	27.53	10.01	14.68	<u>41.16</u>	45.51	43.22
SHOT	<b>46.79</b>	<b>100.00</b>	<b>63.75</b>	27.47	55.25	36.70	34.00	75.94	46.97
TTAC	<u>42.60</u>	94.52	<u>58.73</u>	30.43	72.11	42.80	31.59	74.07	44.29
OURS	41.40	<b>100.00</b>	58.56	<b>38.86</b>	<u>93.35</u>	<b>54.87</b>	<b>38.60</b>	<b>98.06</b>	<b>55.40</b>

## 4.5. Competing Methods

As no existing TTT methods have considered the challenges in open-world TTT, we benchmark the following test-time training methods. **TEST** directly evaluates the source domain model on testing data. **BN** [19] update batch norm statistics on the testing data for test-time adaptation. **TTT++** [27] aligns source and target domain distribution by minimizing the F-norm between the mean covariance. **TENT** [43] updates model batch norm affine parameters by minimizing the entropy loss on testing data. **SHOT** [25] implements test-time training by entropy minimization and self-training. SHOT assumes the target domain is class balanced and introduced an entropy loss to encourage uniform distribution of the prediction results. **TTAC** [39] employs distribution alignment at both global and category levels to facilitate test-time training. Finally, we present our method (**Ours**) which combines self-training with prototype expansion to accommodate the strong OOD samples. For all competing methods, by default, we equip them with the same strong OOD detector introduced in Sect. 3.2.

## 4.6. Evaluation of Open-World TTT

### Open-World TTT on Noise Corrupted Target Domain:

We first evaluate open-world test-time training under noise corrupted target domain. We treat CIFAR10, CI-

FAR100 [20] and ImageNet [9] as the source domain and test-time adapt to CIFAR10-C, CIFAR100-C, and ImageNet-C as the target domain respectively. For experiments on CIFAR10/100, we introduce random noise, MNIST, SVHN, Tiny-ImageNet with non-overlap categories, and CIFAR100 as strong OOD testing samples. For ImageNet-C, we introduce random noise, MNIST, and SVHN as strong OOD samples. The results are presented in Tab. 1, Tab. 2 and Tab. 3 respectively. We make the following observations from the results. First, direct testing does not necessarily gives the worst performance when strong OOD samples are drawn from random noise distribution. This suggests the impact of strong OOD samples is more severe than some corruptions on target domain testing data. Second, BN and TTAC perform relatively well compared with TENT and SHOT across different types of strong OOD testing samples. This result indicates that test-time adaptation through distribution alignment is in general more robust than self-training based methods under the interference of strong OOD testing samples. Finally, our proposed method consistently outperforms all competing methods under most experiment settings, suggesting the effectiveness of the proposed method. In addition to vertically comparing different competing methods, we further notice that the performance variation among different methods is significantly higher on using random noise and MNIST as strong OOD samples than using Tiny ImageNet and CIFAR100 as strong OOD samples. We owe the difference to the fact that there is a less significant distribution shift between CIFAR10 and CIFAR100 than between CIFAR10 and random noise/MNIST.

**Open-World TTT on Style Transfer Target Domain:** We further evaluate VisDA-C and ImageNet-R to demonstrate the effectiveness of the style transfer target domain. For both ImageNet-R and VisDA-C the strong OOD samples

Table 4: Test time training results on ImageNet-R.

Method	noise			MNIST			SVHN		
	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
TEST	35.83	<b>100.00</b>	52.76	35.50	99.96	52.39	35.75	<b>98.32</b>	<b>52.43</b>
BN	40.05	<b>100.00</b>	57.19	39.86	<b>99.98</b>	57.00	<b>38.16</b>	69.41	49.25
TENT	32.35	94.96	48.26	18.77	31.70	23.58	29.17	65.12	40.29
SHOT	39.20	<b>100.00</b>	56.32	37.30	96.30	53.77	32.37	57.53	41.43
TTAC	<u>41.24</u>	<b>100.00</b>	<u>58.40</u>	<u>40.55</u>	99.41	<u>57.60</u>	36.68	68.48	47.77
Paper	<b>41.66</b>	<b>100.00</b>	<b>58.82</b>	<b>41.40</b>	<b>100.00</b>	<b>58.56</b>	<b>41.47</b>	<b>99.90</b>	<b>58.61</b>

Table 5: Test time training results on VisDA-C.

Method	noise			MNIST			SVHN		
	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
TEST	48.22	<b>100.00</b>	65.07	48.59	<b>98.80</b>	<u>65.14</u>	49.38	<b>98.48</b>	64.95
BN	52.58	<b>100.00</b>	68.92	50.11	82.34	62.30	<u>58.49</u>	<u>86.19</u>	<b>69.69</b>
TENT	58.80	98.99	73.78	46.27	68.19	55.13	51.00	81.48	62.73
SHOT	<u>62.09</u>	<b>100.00</b>	<u>76.61</u>	48.19	36.00	41.21	56.47	62.84	59.48
TTAC	59.21	<b>100.00</b>	74.38	<u>52.04</u>	67.38	58.72	57.94	66.27	61.83
OURS	<b>63.93</b>	<b>100.00</b>	<b>78.00</b>	<b>64.18</b>	<u>90.96</u>	<b>75.26</b>	<b>64.76</b>	74.80	<u>69.42</u>

are selected in a similar way to ImageNet-C and the results are presented in Tab. 4 and Tab. 5. We draw similar conclusions from the results as with common corruptions. Our proposed method consistently outperforms competing methods.

#### 4.7. Ablation Study

We investigate the effectiveness of the components in the proposed method. In particular, we evaluate the effectiveness of prototype clustering (Pro. Cls.), OOD detection (OOD Det.), prototype expansion (Pro. Exp.), and finally distribution alignment regularization (Dist. Al.). We report TTT performance as  $Acc_H$  on CIFAR10-C as weak OOD and MNIST, SVHN, and CIFAR100-C as strong OOD in Tab. 6. When strong OOD detection (O.D.) is not present, all testing samples are classified as source domain classes, thus resulting in  $Acc_N = 0\%$ . When strong OOD detector (O.D.) is enabled for inference, we observe a drop of source domain accuracy  $Acc_S$  with significantly improved strong OOD accuracy  $Acc_N$ . We further incorporate prototype clustering (P.C.) on the target domain by updating source domain prototypes  $\mathcal{P}_s$  only. As no strong OOD prototypes are present, direct self-training with source domain prototypes does not necessarily improve  $Acc_S$  and  $Acc_N$  simultaneously. Therefore, as we include prototype expansion (P.E) to dynamically expand strong OOD prototypes, consistent improvements upon O.D. without adaptation are observed on the three datasets. Finally, as we combine distribution alignment (D.A) to regularize self-training, we achieve the best performance on all categories of accuracies. The combined approach also significantly outperforms distribution alignment with strong OOD detection alone.

#### 4.8. Additional Analysis

In this section, we provide additional analysis of open-world TTT from several aspects and explore alternative de-

Table 6: Ablation study on CIFAR10-C as weak OOD samples. We investigate the effectiveness of OOD Detection (O.D.), Prototype Clustering (P.C.), Prototype Expansion (P.E.) and Distribution Alignment (D.A.).

O.D.	P.C.	P.E.	D.A.	Noise			SVHN			CIFAR100-C		
				$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$	$Acc_S$	$Acc_N$	$Acc_H$
-	-	-	-	70.6	0.0	0.0	70.6	0.0	0.0	70.6	0.0	0.0
✓	-	-	-	68.6	100.0	81.4	60.9	86.4	71.5	52.7	74.2	61.7
✓	$\mathcal{P}_s$	-	-	65.2	91.5	76.1	60.9	90.0	72.7	56.3	69.0	62.0
✓	$\mathcal{P}_s + \mathcal{P}_n$	✓	-	68.7	99.8	81.4	65.3	95.0	77.4	52.6	78.9	63.2
✓	-	-	✓	72.9	88.8	80.1	78.1	88.0	82.8	70.5	78.7	74.4
✓	$\mathcal{P}_s + \mathcal{P}_n$	✓	✓	85.5	98.6	91.6	85.0	87.9	86.4	74.1	84.6	79.0

Table 7: Evaluating the robustness of our method on CIFAR10-C under different ratios of strong to weak OOD samples.

Ratio	noise	MNIST	SVHN	Tiny-ImageNet	CIFAR100-C
0.2	90.83	89.59	86.91	77.64	69.15
0.4	91.59	90.55	88.51	77.67	76.99
0.6	91.80	90.92	88.28	77.68	78.77
0.8	91.70	90.19	86.65	77.42	79.58
1.0	91.34	90.34	86.91	77.70	79.04

signs of the model. The performance of open-world TTT is also investigated under different testing data compositions.

**Cumulative performance in the TTT process:** We first present the cumulative testing performance under open-world TTT protocol in Fig. 4. We report  $Acc_H$  for multiple TTT methods. Our method consistently outperforms the competing methods during the whole TTT procedure, and the performance gets progressively better as the number of test samples increases.

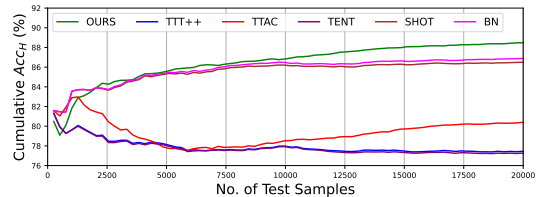


Figure 4: Comparison of test-time cumulative  $Acc_H$ .

**Performance under different open world data ratios:** In real-world applications, the ratio of strong OOD to weak OOD in open-world test data is variable. We examine the impact of different ratios between weak and strong OOD samples on open-world TTT performance. Specifically, we control the ratio between strong to weak OOD samples from 0.2 to 1.0.  $Acc_H$  on CIFAR10-C are presented in Tab. 7. Experiments show that our approach is not sensitive to the data ratio and can be applied to a variety of data ratio scenarios.

**Comparison with fixed thresholds:** We use different fixed thresholds  $\tau_f$  to divide the visible and invisible classes and compare it with the performance of our adaptive threshold  $\tau^*$ . As shown in Fig. 5, it is impossible to find a fixed



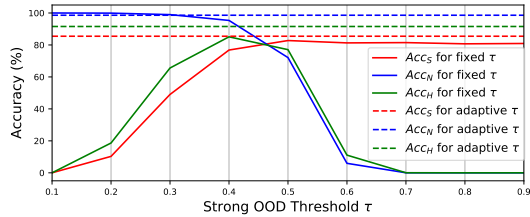


Figure 5: Comparison between the optimal strong OOD threshold  $\tau^*$  (dotted lines) and fixed threshold  $\tau$  (solid lines) range from 0.1 to 0.9.

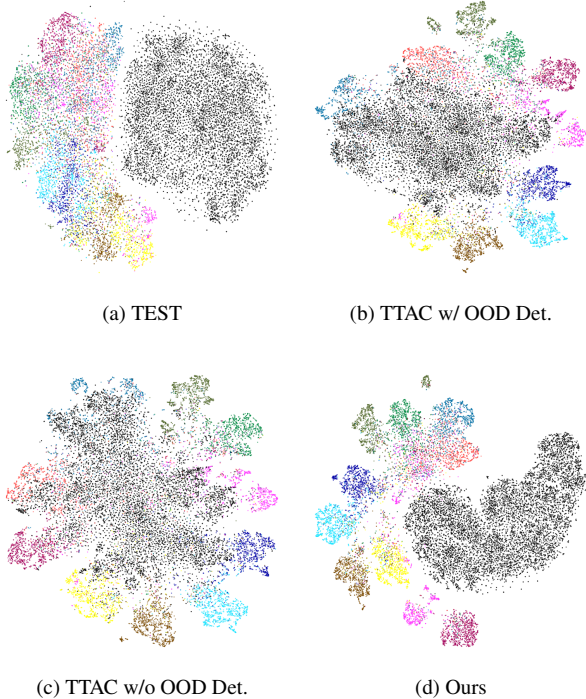


Figure 6: T-SNE visualization on Cifar10-C test set with SVHN as strong OOD samples. Black dots are strong OOD examples while other colors are weak OOD examples. The colors indicate different classes.

threshold that outperforms the adaptive threshold for strong OOD detection.

**T-SNE Visualization** To demonstrate the representation adaptation effect of different methods, we use t-SNE[42] to reduce the dimensionality of the features for visualization. In Fig. 6, we compared the features adapted by TEST (without adaptation), TTAC [39], and our method. We observe a more distinct separation between weak (colorful points) and strong (black points) OOD sample features learned by our method, compared to without adaptation and state-of-the-art TTT method. We also observe that filtering out strong OOD samples (w/ OOD Det.) benefits the generic TTT method, e.g. (b) TTAC w/ OOD Det. yields more distinguishable separation between weak and strong OOD samples of weak OOD and strong OOD features.

Table 8: Performance under mixed strong OOD samples.

Metrics	TEST	BN	TENT	SHOT	TTT++	TTAC	Ours
$Acc_S$	63.90	75.18	58.10	76.63	60.58	<b>85.01</b>	82.19
$Acc_N$	88.20	90.37	77.19	60.94	87.39	66.78	<b>96.76</b>
$Acc_H$	74.11	82.08	66.30	67.89	71.56	74.80	<b>88.88</b>

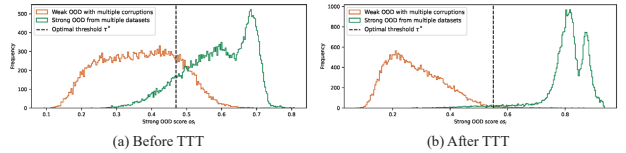


Figure 7: Noisy bimodal distribution under multiple corruption.

### Performance Under Multiple Corruption Datasets

We provide an evaluation, in Tab. 8, of combining “snow”, “contrast”, and “glass blur” on CIFAR10-C as weak OOD and combining random noise, MNIST, and SVHN as strong OOD. It is evident that our approach consistently outperforms all competitors under both mixed weak OOD samples and mixed strong OOD samples.

We illustrate the OOD score distributions in Fig. 7 for the experiment with mixed weak OOD samples and mixed strong OOD samples presented in Tab. 8. In Fig. 7 (a) and (b) we respectively show the OOD score distributions before and after test-time training. We conclude that, although the more challenging mixed setting makes it harder to differentiate weak and strong OOD samples before TTT, our method can well separate these two groups after TTT. This suggests the robustness of our method under noisy OOD score distributions.

## 5. Conclusion

Test-time training has been extensively studied to enable adaptation to unknown target distribution with low inference latency. In this work, we investigate the robustness of test-time training when testing data is contaminated with strong OOD samples, a.k.a. open-world test-time training (OWTTT). We proposed a hyper-parameter-free strong OOD detector that benefits both self-training and distribution alignment for OWTTT. We further allow the prototype pool to expand dynamically with which self-training can allow better separation between weak and strong OOD samples. Extensive evaluations on five OWTTT benchmarks demonstrated the effectiveness of the proposed method.

**Acknowledgments.** This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62106078, the A\*STAR under Project C210112059 and M23L7b0021, Sichuan Science and Technology Program under Project 2023NSFSC1421, Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.: 2017ZT07X183) and Guangdong R&D key project of China (No.: 2019B010155001).

## References

- [1] David J Aldous, Ildar A Ibragimov, Jean Jacod, and David J Aldous. *Exchangeability and related topics*. 1985.
- [2] Kelsey Allen, Evan Shelhamer, Hanul Shin, and Joshua Tenenbaum. Infinite mixture prototypes for few-shot learning. In *International Conference on Machine Learning*, 2019.
- [3] Eric Arazo, Diego Ortego, Paul Albert, Noel E O’Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *International Joint Conference on Neural Networks*, 2020.
- [4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 2010.
- [5] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [6] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 295–305, 2022.
- [7] Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7181–7190, 2022.
- [8] Francesco Croce, Sven Gowal, Thomas Brunner, Evan Shelhamer, Matthias Hein, and Taylan Cemgil. Evaluating the adversarial robustness of adaptive test-time defenses. In *International Conference on Machine Learning*, 2022.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.
- [10] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [11] Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE Transactions on Neural Networks*, 2019.
- [12] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015.
- [13] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- [14] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. In *Advances in Neural Information Processing Systems*, 2022.
- [15] Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J Zico Kolter. Test time adaptation via conjugate pseudo-labels. In *Advances in Neural Information Processing Systems*, 2022.
- [16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [17] 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 International Conference on Computer Vision*, pages 8340–8349, 2021.
- [18] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.
- [19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.
- [20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [21] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *International Conference on Machine Learning*, 2012.
- [22] Qicheng Lao, Xiang Jiang, and Mohammad Havaei. Hypothesis disparity regularized mutual information maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8243–8251, 2021.
- [23] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.
- [24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [25] 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*, 2020.
- [26] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation. In *Advances in Neural Information Processing Systems*, 2021.
- [27] 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? In *Advances in Neural Information Processing Systems*, 2021.
- [28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bisacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [29] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofu Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, 2022.
- [30] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiqian Wen, Yaofu Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.
- [31] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

- [32] Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 2014.
- [33] Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. In *International Joint Conference on Artificial Intelligence*, 2021.
- [34] Joaquin Quiñero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.
- [35] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.
- [36] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *European Conference on Computer Vision*, 2018.
- [37] Walter J. Scheirer, A. de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [38] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 2020.
- [39] Yongyi Su, Xun Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. In *Advances in Neural Information Processing Systems*, 2022.
- [40] Yu Sun, Xiaolong Wang, Zhuang Liu, John J. Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2019.
- [41] Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5940–5947, 2020.
- [42] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [43] 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.
- [44] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 2018.
- [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] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6502–6509, 2020.
- [47] Yiming Xu, Lin Chen, Lixin Duan, Ivor W. Tsang, and Jiebo Luo. Open set domain adaptation with soft unknown-class rejection. *IEEE Transactions on Neural Networks*, 2021.