Dynamic Against Dynamic: An Open-set Self-learning Framework

Haifeng Yang[1,2*], **Chuanxing Geng**[1,2,3*], **Pong C. Yuen**[3] and **Songcan Chen**[1,2†]
[1]Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
[3]Hong Kong Baptist University
{SZ2216008, gengchuanxing, s.chen}@nuaa.edu.cn, pcyuen@comp.hkbu.edu.hk

Abstract

In open-set recognition, existing methods generally learn statically fixed decision boundaries using known classes to reject unknown classes. Though they have achieved promising results, such decision boundaries are evidently insufficient for universal unknown classes in dynamic and open scenarios as they can potentially appear at any position in the feature space. Moreover, these methods just simply reject unknown class samples during testing without any effective utilization for them. In fact, such samples completely can constitute the true instantiated representation of the unknown classes to further enhance the model's performance. To address these issues, this paper proposes a novel dynamic against dynamic idea, i.e., dynamic method against dynamic changing open-set world, where an open-set self-learning (OSSL) framework is correspondingly developed. OSSL starts with a good closed-set classifier trained by known classes and utilizes available test samples for model adaptation during testing, thus gaining the adaptability to changing data distributions. In particular, a novel self-matching module is designed for OSSL, which can achieve the adaptation in automatically identifying known class samples while rejecting unknown class samples which are further utilized to enhance the discriminability of the model as the instantiated representation of unknown classes. Our method establishes new performance milestones respectively in almost all standard and cross-data benchmarks.

1 Introduction

Traditional supervised classification has achieved great success in past decades, partly due to the closed-set assumption that the training and test samples share the same feature and label spaces [Fang *et al.*, 2021; Zhou, 2022]. However, this assumption does not always hold in the real-world, since we usually know nothing about the incoming test samples, meaning they could potentially come from unknown classes unseen during training, thus drastically weakening the model performance. To meet this challenge, open-set recognition (OSR) is proposed [Scheirer *et al.*, 2013], which aims to construct the

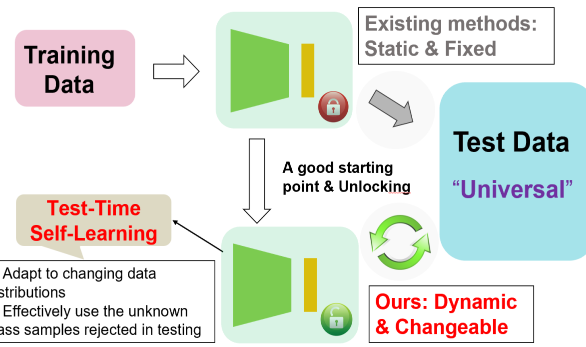


Figure 1: Differences between existing OSR methods and ours. Existing methods generally learn static and fixed classifiers using known class samples for universal unknown classes, whereas ours follows the dynamic against dynamic idea, i.e., dynamic methods against dynamic changing open-set world, where the learned classifier is dynamic and changeable.

models that not only accurately identify known class samples but also effectively reject unknown class samples.

With the unremitting efforts of the researchers, OSR has achieved significant progress [Neal *et al.*, 2018; Oza and Patel, 2019; Chen *et al.*, 2021; Fang *et al.*, 2021; Guo *et al.*, 2021; Kong and Ramanan, 2021; Zhou *et al.*, 2021; Xu *et al.*, 2023]. For example, [Chen *et al.*, 2021] introduced the reciprocal point technique to model the unexploited extra-class space while [Zhou *et al.*, 2021] adopted the placeholder technique reserve space for unknown classes, thus gaining the more accurate decision boundaries. [Xu *et al.*, 2023] recently introduced the popular supervised contrastive learning technique to aim at learning more effective representations for OSR and established a new performance milestone.

Despite the encouraging advance, the ability of existing methods to deal with agnostic unknown classes actually remains limited due to the following issues:

✓ **Static and Fixed Decision Boundaries**. Existing methods generally learn statically fixed decision boundaries using known class samples (as shown in Figure 1) for the universal unknown classes in dynamic and open scenarios, which is evidently insufficient as these classes can emerge unexpectedly at any position in the feature space.

✓ **Knowledge Waste of Rejected Test Samples**. Existing methods just simply reject unknown class samples without any additional operations to effectively utilize them. In fact, such samples completely can constitute the true instantiated representation of the unknown class space to further enhance the model's performance.

To address these unsolved issues in open-set recognition, this paper proposes a novel *dynamic against dynamic* idea, i.e., *dynamic methods against dynamic changing open-set world*. Therefore, an open-set self-learning (OSSL) framework is correspondingly developed, which starts with a good closed-set classifier trained by known classes and then self-trains with the available tested yet commonly deprecated samples for the model adaptation during testing. Specifically, OSSL first leverages the well-trained closed-set classifier to perform the identity inference of all the test samples, where it divides the entire test set into three subsets according to the logit scores of samples obtained by the closed-set classifier. Then these subsets will be submitted to a novel self-matching module which adaptively implements the classifier update. Afterwards, the updated classifier serves as the new inferrer, repeating the process until the model converges. In summary, our contributions can be highlighted as follows:

- To our best knowledge, it is the first time that the dynamic against dynamic idea, i.e., dynamic methods against dynamic changing open-set world, is introduced in the modeling of the OSR problem, where an open-set self-learning framework is correspondingly developed.
- A novel self-matching module is design for OSSL, which can achieve the adaptation in automatically identifying known class samples while rejecting unknown samples which are further utilized to enhance the discriminability of the model as the true instantiated representation of the unknown class space.
- Extensive experiments verify the effectiveness of our OSSL, establishing new performance milestones respectively in almost all standard and cross-data benchmark datasets.

2 Related Work

2.1 Open Set Recognition

The OSR problem is initially formalized in [Scheirer *et al.*, 2013] and had achieved notable advancements in the past decade, where numerous OSR methods have been developed, which can be mainly divided into two categories: discriminative-based and generative-based methods [Geng *et al.*, 2021]. **Discriminative-based methods** aim to accurately model known classes using various techniques in order to effectively reject unknown classes [Geng *et al.*, 2020; Chen *et al.*, 2020; Chen *et al.*, 2021; Zhou *et al.*, 2021; Huang *et al.*, 2022; Wang *et al.*, 2022; Cevikalp *et al.*, 2023; Xu *et al.*, 2023]. **Generative-based methods** usually employ adversarial generation techniques to generate pseudo-unknown class samples, then transforming the original K-class OSR problem into a $K+1$-class closed-set multi-class classification problem [Neal *et al.*, 2018; Perera and Patel, 2021; Kong and Ramanan, 2021; Geng and Chen, 2022].

As mentioned previously, though these methods have shown promising results, their decision boundaries are all static and fixed, which evidently confines the further performance improvement when facing dynamic and open scenarios. This motivates us to propose the dynamic against dynamic idea, constructing dynamic changeable models to face the dynamic open-set world.

2.2 Self-Training

Self-training [Scudder, 1965; Lee, 2013] is a commonly used semi-supervised learning paradigm that allows the model to leverage the abundant unlabeled data to enhance its learning capabilities. The main focus of self-training lies in that limited labeled data often make the learned model unreliable, resulting in significant noise in pseudo-labeled data, which further induces the model to update in the wrong direction. To address this issue, large numbers of methods have been proposed [Zou *et al.*, 2019; Mukherjee and Awadallah, 2020; Xie *et al.*, 2020; Wei *et al.*, 2021; Karisani, 2023; Garg *et al.*, 2023]. For example, [Zou *et al.*, 2019] proposed a confidence regularized self-training framework, treating pseudo-labels as continuous latent variables jointly optimized by alternating optimization. Recently, some researchers have attempted to extend self-training to open scenarios, i.e., there are new classes in unlabeled data that do not appear in labeled data [Cao *et al.*, 2022; Zhuang *et al.*, 2024].

Similarly, we also explore the application of self-traininig in open-set recognition. Additionally, we have relatively sufficient labeled known class samples here, which to some extent ensures the reliability of the initial classifier inference.

2.3 Test-Time Adaptation

Test-time adaptation (TTA) aims to improve the model performance through model adaptation with the changing test data distributions. One pipeline of this study is to jointly train a source model via both supervised and self-supervised objectives, then adapting the model through self-supervised objectives during testing [Sun *et al.*, 2020; Liu *et al.*, 2021; Bartler *et al.*, 2022]. The other pipeline is the full test-time adaptation method, which adapts a model with only test data [Schneider *et al.*, 2020; Wang *et al.*, 2020; Zhang *et al.*, 2022; Iwasawa and Matsuo, 2021; Niu *et al.*, 2022]. For example, [Wang *et al.*, 2020] proposed to adapt by test entropy minimization while [Zhang *et al.*, 2022] adopted the prediction consistency maximization over different augmentations.

Our work falls into the latter pipeline. In particular, to our best knowledge, it is the first time to apply the TTA paradigm to the OSR scenario, further enriching the research on TTA.

3 Methodology

3.1 Preliminary

Let $\mathcal{D}_{tr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{tr}}$ denote the training set, where $\mathbf{x}_i$ is the $i-$th sample and $y_i \in \mathcal{C}_{tr} = \{1, 2, 3, ..., K\}$ represents the corresponding known class label. Similarly, let $\mathcal{D}_{te} = \{(\mathbf{x}_j, y_j)\}_{j=1}^{N_{te}}$ be the test set, where $y_i \in \mathcal{C}_{te} = \{1, 2, 3, .., K, K+1\}$. Note that $K+1$ demonstrates a group of unknown classes, which may contain more than one class

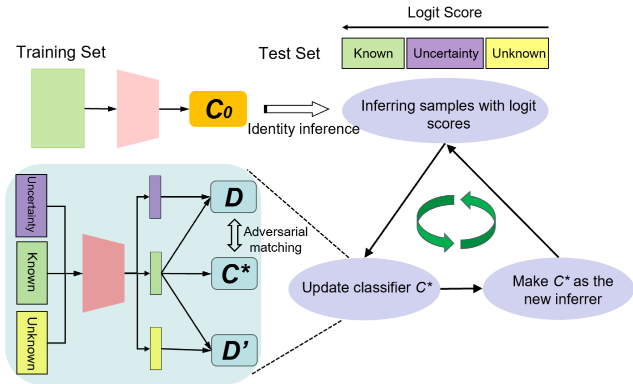


Figure 2: The overview of our open-set self-learning framework.

[Zhou *et al.*, 2021]. Open-set recognition aims to learn a robust classifier that only accurately classify the known classes but also effectively reject unknown classes.

3.2 Open-Set Self-learning Framework

To face the challenge from the universal unknown classes in dynamic and open scenario, we propose the dynamic against dynamic idea and develop an open-set self-learning (OSSL) framework, which consists of two key components: one is a well-trained closed-set classifier as its starting point, and the other is a novel self-matching module. The former aims to ensure the reliability of pseudo labeling of test data to some extent, while the latter aims to achieve the model adaptation on test data. In specific, OSSL first leverages the well-trained closed-set classifier to perform the identity inference of all the test samples, where it divides the entire test set into three parts, namely known-label set, uncertainty set and unknown-label set according to the logit scores of samples obtained by the closed-set classifier. Then these data sets will be submitted to a novel self-matching module, which contains three parts: a classifier part C^* applied to the known-label set, an adversarial matching part D applied to the known-label and uncertainty sets, and a detection part D' applied to the known-label and unknown-label sets. These parts work collaboratively to facilitate the enhancement and update of the classifier C^*. Afterwards, the updated classifier serves as the new inferrer, repeating the process until the model converges. Figure 2 shows the overview of our OSSL framework. Next, we will elaborate on these two key components of OSSL, i.e., the good starting point and the self-matching module.

The Good Starting Point of OSSL

The core of this component is that the inference of initial classifier should be reliable to some extent, especially for unknown classes. Fortunately, the recent study [Vaze *et al.*, 2022] has shown that a well-trained closed-set classifier can achieve this, where they just train a closed-set classifier using various tricks such as data augmentation, label smoothing, etc, achieving at least comparable performance to SOTA baselines [Chen *et al.*, 2021; Xu *et al.*, 2023]. Therefore, we here employ the network architecture in [Vaze *et al.*, 2022] as our model's backbone, the trained closed-set classifier as our initial classifier C_0 whose outputs are logit vectors.

Algorithm 1 Training Procedure of OSSL Framework

Input: Data from training set $\mathcal{D}_{tr}$; Data from testing set $\mathcal{D}_{te}$;
Parameter: Starting point classifier $C_0(\cdot)$; Adversarial matching part $D(\cdot)$; Detection part $D'(\cdot)$;

Initialize model parameters $D(\cdot)$, $D'(\cdot)$.
for $epoch$ =1 in $epoch_{max}$ **do**
  Divide $\mathcal{D}_{te}$ based on probability scores from classifier C^* (its initial one is C_0) into three parts: T_1 , T_2 and T_3.
  Sample data from training set to get $Tr_s \subset \mathcal{D}_{tr}$.
  Based on Eq.(6) , using data from Tr_s and known-label set T_1 to calculate $\mathcal{L}'_{C^*}$.
  Obtain data from known-label set T_1, unknown-label set T_3 and Tr_s to compute $\mathcal{L}'_{D'}$ based on Eq.(7) .
  Calculate ω^s and ω^t in the method of Eq.(3) .
  Based on Eq.(8) , utilize calculated ω^s and ω^t to compute ${\mathcal{L}'}_D$ for our adversarial matching part.
  Based Eq.(9) , use data from unknown-label set T_3 to calculate $\mathcal{L}_{Mar}$.
  Calculate the total loss of our OSSL framework based on Eq.(10).
  Based on the loss function, update parameters of our framework.
end for

The role of the initial classifier C_0 is to provide the first identity inference for the test data, where it divides the test set into three parts according the logit scores of samples obtained by C_0, namely, known-label set $T_1 = \{\boldsymbol{x}_j | \max(C_0(\boldsymbol{x}_j)) > \mu\}$, unknown-label set $T_3 = \{\boldsymbol{x}_j | \max(C_0(\boldsymbol{x}_j)) - \min(C_0(\boldsymbol{x}_j)) < \gamma\}$, and the remaining data set, i.e., uncertainty set T_2. Note that considering the characteristic of the unknown class logit vectors, which tends to be uniformly distributed, we here adopt the relative differences of logit values rather than their magnitudes. Then these preliminarily inferred samples will be submitted to the self-matching module of OSSL for further processing.

Remark. In fact, all existing OSR approaches can serve as our initial classifiers. This implies that our OSSL is orthogonal to existing methods and can leverage them as starting points to achieve even better performance.

The Self-Matching Module of OSSL

The self-matching module aims to efficiently update the classifier C^* to adapt to the changing test distribution. In this module, the known-label set T_1 is directly used to update C^*, however, the number of its samples may not be sufficient, just about 28% of the batch-size (256) in our framework. Thus, the adversarial matching part D is proposed to fully utilize the uncertainty set T_2 to update C^*, whose core is a sample-level weighted mechanism, mainly drawing on the idea of distribution alignment in [You *et al.*, 2019]. D aims to maximize the alignment/matching of known-class data distribution in T_1 and T_2 while automatically detecting unknown class samples in T_2. Further, the key to the sample-level weighted mechanism here is to quantify the similarity between each sample and the known class distribution, which can be achieved by the detection part D' as its objective is to predict samples from known class distribution as 1 and samples from unknown class distribution as 0. In summary, these three parts work collaboratively to facilitate the enhancement and update of C^*. Their specific details will be elaborated in the followings.

For the classifier part, it mainly applies the cross-entropy loss to known-label set to gain the classifier update,

$$\mathcal{L}_{C^*} = \mathbb{E}_{\boldsymbol{x}\sim T_1} L(\widehat{\boldsymbol{y}}, \text{softmax}(C^*(F(\boldsymbol{x}))), \tag{1}$$

where $F(\cdot)$ denotes the feature extractor network, C^* is a linear fully connected layer classifier with only one layer, whose initialization parameters come from C_0 and outputs are logit vectors. $\widehat{\boldsymbol{y}}$ denotes the pseudo-known label obtained by the previous classifier (the initial one is C_0).

For the adversarial matching part, it operates on known-label and uncertainty sets, defined as follows:

$$\mathcal{L}_D = -\mathbb{E}_{\boldsymbol{x}\sim T_1}\omega^s \log(D(F(\boldsymbol{x}))) - \mathbb{E}_{\boldsymbol{x}\sim T_2}\omega^t \log(1 - D(F(\boldsymbol{x}))). \tag{2}$$

Its core is the sample-level weighted mechanism, i.e., ω^s and ω^t, which represent the probabilities of corresponding samples belonging to known classes, and their values can be obtained by the following equations,

$$\omega^s(\boldsymbol{x}) = \frac{H(\widehat{\boldsymbol{y}})}{\log|\mathcal{C}_{tr}|} - d', \ \ \omega^t(\boldsymbol{x}) = d' - \frac{H(\widehat{\boldsymbol{y}})}{\log|\mathcal{C}_{tr}|}, \tag{3}$$

where H denotes the entropy of a vector, which is normalized by its maximum value ($log|\mathcal{C}_{tr}|$) so that ω^s and ω^t are restricted into [0, 1]. $|\mathcal{C}_{tr}|$ represents the number of known classes. d' quantifies the similarity of $\boldsymbol{x}$ to known classes and can be achieved by the detection part D' ($d' = D'(F((\boldsymbol{x})))$), defined as follows:

$$\mathcal{L}_{D'} = -\mathbb{E}_{\boldsymbol{x}\sim T_1} \log(D^{'}(F(\boldsymbol{x}))) - \mathbb{E}_{\boldsymbol{x}\sim T_3} \log(1 - D^{'}(F(\boldsymbol{x}))). \tag{4}$$

In summary, the total loss function of the self-matching module can be written as:

$$\begin{aligned} &\max_{D} \min_{F,C^*} \mathcal{L}_{C^*} - \mathcal{L}_D \\ &\min_{D'} \mathcal{L}_{D'} \end{aligned} \tag{5}$$

Some Enhancement Strategies

To further improve the reliability of model inference, we enhance the model from the following two strategies.

Injecting a Small Amount of Ground-Truth Data. We sample a small amount of labeled data of known classes from the training set, i.e. $Tr_s \subset \mathcal{D}_{tr}$, and add it to the self-learning process of the model to enhance the reliability of model inference. As a result, the loss functions of the self-matching module respectively become:

$$\begin{aligned} \mathcal{L}'_{C^*} = &\mathbb{E}_{\boldsymbol{x}\sim T_1} L(\widehat{\boldsymbol{y}}, \text{softmax}(C^*(F(\boldsymbol{x}))) \\ &+ \mathbb{E}_{\boldsymbol{x}\sim Tr_s} L(\boldsymbol{y}, \text{softmax}(C^*(F(\boldsymbol{x}))), \end{aligned} \tag{6}$$

$$\begin{aligned} \mathcal{L}'_{D'} = &- \mathbb{E}_{\boldsymbol{x}\sim T_1 \cup Tr_s} \log(D^{'}(F(\boldsymbol{x}))) \\ &- \mathbb{E}_{\boldsymbol{x}\sim T_3} \log(1 - D^{'}(F(\boldsymbol{x}))), \end{aligned} \tag{7}$$

$$\begin{aligned} \mathcal{L}'_{D} = &- \mathbb{E}_{\boldsymbol{x}\sim T_1 \cup Tr_s}\omega^s \log(D(F(\boldsymbol{x}))) \\ &- \mathbb{E}_{\boldsymbol{x}\sim T_2}\omega^t \log(1 - D(F(\boldsymbol{x}))). \end{aligned} \tag{8}$$

Marginal Logit Loss for Unknown Classes. In fact, we usually expect the logits of unknown class samples to tend towards the uniform distribution. However, such uniformly-distributed logits do not necessarily lead to sufficiently small values of them. To address this issue, we here introduce a marginal logit loss for the unknown-label set T_3 inspired by [Deng *et al.*, 2022], defined as follows,

$$\mathcal{L}_{Mar} = \frac{1}{N_{T_3}} \sum_{i=1}^{N_{T_3}} \sum_{k=1}^{K} max(0, \lambda + v_{i,k}), \tag{9}$$

where $v_{i,k}$ denotes the element value to the corresponding position of the logit vector. Therefore, the ultimately total loss of our OSSL can be summarized as follows:

$$\begin{aligned} &\max_{D} \min_{F,C^*} \mathcal{L}'_{C^*} - \mathcal{L}'_D \\ &\min_{D',F} \mathcal{L}'_{D'} + \mathcal{L}_{Mar}. \end{aligned} \tag{10}$$

The overall training procedure of OSSL is detailed in Algorithm 1.

4 Experiments

4.1 Implementation Details

We employ the network architecture in [Vaze *et al.*, 2022] as the backbone of our learning framework[1], and choose the Stochastic Gradient Descent (SGD) technique as the optimizer. The same data augmentations in [Vaze *et al.*, 2022], like random horizontal flip, random cropping, etc., are also adopted here. Considering that the feature extractor $F(\cdot)$ is a already well-trained network, we here set its learning rate to 10^{-4} for TinyImageNet while 10^{-5} for other datasets. Furthermore, the learning rates of other parts except for $F(\cdot)$ are uniformly set to 0.01. For the threshold parameters used in the partition of test set, we set $\mu = 0.3$, $\gamma = 0.03$ for TinyImageNet, $\mu = 0.5$, $\gamma = 0.03$ for Cifar10, Cifar+10, Cifar+50, while $\mu = 0.8$, $\gamma = 0.02$ for MNIST and SVHN. In addition, the number of samples from Tr_s in each batch is set to 16 (the batch-size is 256), while the hyper-parameter λ in $\mathcal{L}_{Mar}$ is set to 2 for all benchmark datasets.

4.2 Experiments in Standard-Dataset Benchmarks

Datastes

We here follow the protocol defined in [Neal *et al.*, 2018], and provide six standard OSR benchmarks:

- **MNIST, SVHN, Cifar10**. For MNIST [Lake *et al.*, 2015], SVHN [Netzer *et al.*, 2011], and CIFAR10 [Krizhevsky, 2009], each dataset comprises 10 distinct classes, in which 6 classes are randomly selected as known classes and the other 4 classes as unknown.
- **Cifar+10, Cifar+50**. In this series of experiments, we select 4 classes from the CIFAR10 dataset to serve as known classes for training and 10 or 50 classes from the CIFAR100 dataset as unknown classes in specific experiment.

[1] https://github.com/ChuanxingGeng/OSSL

Method	MNIST	SVHN	Cifar10	Cifar+10	Cifar+50	TinyImageNet
Softmax	0.978	0.886	0.677	0.816	0.805	0.577
OpenMax (CVPR'2016)	0.981	0.894	0.654	0.817	0.795	0.576
OSRCI (ECCV'2018)	0.988	0.910	0.699	0.838	0.827	0.586
CROSR (CVPR'2019)	0.991	0.899	0.883	0.912	0.905	0.589
C2AE (CVPR'2019)	0.989	0.922	0.895	0.955	0.937	0.748
OpenHybrid (ECCV'2020)	0.995	0.947	0.950	0.962	0.955	0.793
PROSER (CVPR'2021)	-	0.943	0.891	0.960	0.953	0.693
EGT (ICME'2021)	-	0.958	0.821	0.937	0.930	0.709
ARPL (TPAMI'2021)	0.996	0.963	0.901	0.965	0.943	0.762
ARPL+CS (TPAMI'2021)	0.997	0.967	0.910	0.971	0.951	0.782
DIAS (ECCV'2022)	0.992	0.943	0.850	0.920	0.916	0.731
PMAL (AAAI'2022)	0.997	0.970	0.951	0.978	0.969	0.831
ALL_U_NEED (ICLR'2022)	0.993	0.971	0.936	0.979	0.965	0.830
ConOSR (AAAI'2023)	0.997	**0.991**	0.942	0.981	0.973	0.809
ARPL+Ours	0.997 (**+0.1%**)	0.969 (**+0.6%**)	0.919 (**+1.8%**)	0.972 (**+0.7%**)	0.951 (**+0.8%**)	0.772 (**+1.0%**)
OSSL (Ours)	**0.998**	0.976	**0.952**	**0.988**	**0.980**	**0.842**

Table 1: Evaluation on open-set detection (AUROC) under the standard-dataset setting. APRL+Ours means that the classifier in ARPL is chosen as the starting point classifier. All of the reported results displayed are averaged across the same five distinct splits.

- **TinyImageNet**. TinyImageNet, a derived subset of the larger ImageNet [Russakovsky *et al.*, 2014] dataset, encompasses 200 classes, where 20 classes are selected as known classes, while the remaining 180 classes as unknown classes.

Evaluation Metrics

Following the mainstream works [Geng *et al.*, 2021], the area under ROC curve (AUROC) is used to evaluate the model's ability to detect unknown classes shown in Table 1, while the accuracy (ACC) is used to evaluate the performance of closed-set classification detailed in the supplementary materials.

Results Comparison

We compare OSSL with 14 classical and leading OSR methods including Softmax, OpenMax [Bendale and Boult, 2015], OSRCI [Neal *et al.*, 2018], CROSR [Yoshihashi *et al.*, 2018], C2AE [Oza and Patel, 2019], OpenHybrid [Zhang *et al.*, 2020], PROSER [Zhou *et al.*, 2021], EGT [Perera and Patel, 2021], ARPL [Chen *et al.*, 2021], DIAS [Moon *et al.*, 2022], PMAL [Lu *et al.*, 2022] , All_U_NEED [Vaze *et al.*, 2022] and [Xu *et al.*, 2023] ConOSR.

Table 1 reports the AUROC results. Thanks to the self-matching module in testing endowing the model with the adaptability to the changing data distributions, our OSSL establishes new performance milestone on almost all benchmark datasets. In particular, in some relatively difficult classification tasks, like TinyImageNet, it significantly outperforms the current two SOTA baselines, i.e., ALL_U_NEED [Vaze *et al.*, 2022] and ConOSR [Xu *et al.*, 2023], respectively achieving performance improvements of 1.2 and 3.3 percentage points.

To further demonstrate the effectiveness of our learning framework, we also employ a popular OSR method, i.e., ARPL [Chen *et al.*, 2021] as the starting point of our OSSL (APRL+Ours). As shown in Table 1, APRL+Ours achieve performance gains across all benchmark datasets. For instance, it achieves an average improvement of 0.35% on MNIST and SVHN whose performance is almost saturated for APRL, while an average improvement of about 1.1% on the remaining benchmark datasets. This also implies that our OSSL is orthogonal to existing methods and can assist them in breaking through existing limitations, achieving further performance improvement.

4.3 Experiments in Cross-Dataset Benchmarks

In this series of experiments, we adhere to the established protocol in [Yoshihashi *et al.*, 2018] to execute our experiments. Within this protocol, all the classes designated for training in the original dataset are employed as in-distribution (ID) data. Meanwhile, samples from an auxiliary dataset are incorporated into the test set, serving as out-of-distribution (OOD) data.

Datasets

- **ID: MNIST; OOD: Omniglot, MNIST-Noise, Noise**. MNIST is selected as ID data, while samples from Omniglot, MNIST-Noise, Noise are selected as OOD data. In specific, the Omniglot dataset consists of a diverse collection of hand-written characters from various alphabets. The Noise dataset is synthesized with each image generated by sampling pixel values from a uniform distribution in the range [0, 1]. Additionally, the MNIST-Noise dataset is created by superimposing the test images from the MNIST dataset onto the Noise images. The OOD data is quantified at 10,000 instances, which equals to the number of test samples in MNIST.
- **ID: Cifar10; OOD: ImageNet, LSUN**. Similarly, Cifar10 here is selected as ID data, while samples from ImageNet and LSUN are selected as OOD data. Since the image size of ImageNet and LSUN [Yu *et al.*, 2015] is different from Cifar10, two different ways, i.e., crop

Method	Omniglot	MNIST-Noise	Noise
Softmax	59.5	64.1	82.9
OpenMax	68.0	72.0	82.6
CROSR	79.3	82.7	82.6
PROSER	86.2	87.4	88.2
ConOSR	95.4	98.7	98.8
OSSL(Ours)	**97.6 (+2.2)**	**99.7 (+1.0)**	**99.2 (+0.2)**

Table 2: Macro-F1 score (%) of different methods under the cross-dataset setting (MNIST as the ID data).

Method	ImageNet		LSUN	
	Crop	Resize	Crop	Resize
Softmax	63.9	65.3	64.2	64.7
OpenMax	66.0	68.4	65.7	66.8
OSRCI	63.6	63.5	65.0	64.8
CROSR	72.1	73.5	72.0	74.9
GFROSR	75.7	79.2	75.1	80.5
PROSER	84.9	82.4	86.7	85.6
ConOSR	89.1	84.3	91.2	88.1
OSSL(Ours)	**96.5 (+7.4)**	**90.8 (+6.5)**	**96.4 (+5.2)**	**95.5 (+7.4)**

Table 3: Macro-F1 score (%) of different methods under the cross-dataset setting (Cifar10 as the ID data).

and resize operations are used to adjust their image sizes to match those in Cifar10. In addition, the numbers of test samples for Cifar10, ImageNet and LSUN are all $10,000$.

Evaluation Metrics

The performance evaluation is conducted using the macro-averaged F1-score (%), which provides a comprehensive measure of the model's effectiveness across the diverse class types.

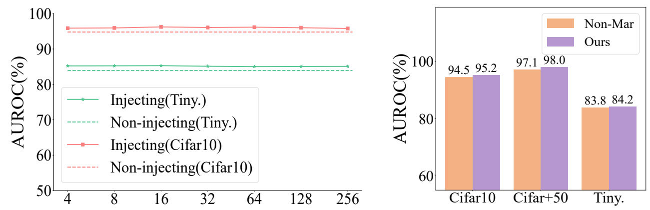
Results Comparison

We compare our OSSL with 7 classical and leading methods. Table 2 and Table 3 respectively report the results on the MNIST and Cifar10 tasks. Compared with the SOTA baseline, i.e., ConOSR [Xu *et al.*, 2023], our OSSL creates new performance milestones with significant advantages on all cross-dataset benchmarks. In particular, for the relative challenging Cifar10 task, it beats ConOSR with an average performance gain of about 6.6%, while an average performance gain of about 1.2% for the MNIST task. This once again proves the effectiveness of such an dynamic changeable learning framework for changing data distributions.

4.4 Effectiveness of Different Enhancements

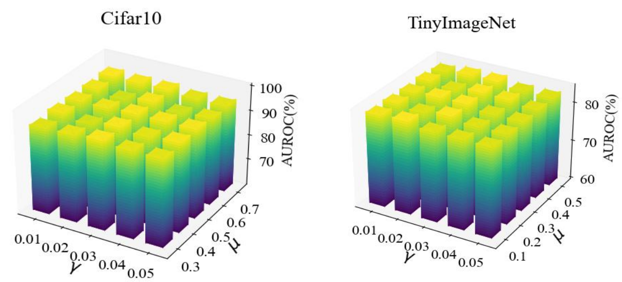
Injecting a Small Amount of Ground-Truth Data

To verify the effectiveness of injecting a small amount of labeled samples, we here conduct the experiments on Cifar10 and TinyImageNet via changing the number of labeled samples from the training set in each batch (the batch size is 256 in all experiments here) during test-time training. Figure 3(a) shows the results. **Our first observation** is that whether injecting a small number of labeled samples to the Cifar10 task



(a) (b)

Figure 3: Experiments on the effectiveness of different enhancement strategies.



(a) (b)

Figure 4: Threshold parameter sensitivity analysis on Cifar10 and TinyImageNet, where (a) reports the results for Cifar10 while (b) for TinyImageNet.

or the TinyImageNet task, the corresponding performances are significantly improved with about 0.4% gains for Cifar10 while about 0.35% gains for TinyImageNet. This verifies that adding a small number of labeled samples can improve the inference performance of the model. **Our second observation** is that as the number of labeled samples increases, there is no obvious trend of performance growth, which can be mainly attributed to the fact that the initial classifier has been well-trained on these samples. Please note that this may not necessarily be a bad thing. In fact, it provides evidence (at least to some extent) in such a scenario where we only obtain a well-trained black-box model and a small amount of authorized training samples, we can still train a model that performs comparably to models trained with large numbers of authorized training samples.

Marginal Logit Loss for Unknown Classes

To verify the effectiveness of the introduced marginal logit loss for unknown classes, we also conduct the relevant experiments. Due to the limited space, we here only present the results on Cifar10, Cifar+50, and TinyImageNet (Figure 3(b)), while the results on other datasets can be found in the supplementary materials. As shown in Figure 3(b), the introduce of $\mathcal{L}_{Mar}$ enables the model to achieve consistent performance improvement on all benchmark datasets.

4.5 Threshold Parameter Sensitivity Analysis

In this section, we undertake a specific sensitivity analysis focusing on the threshold parameter associated with the par-

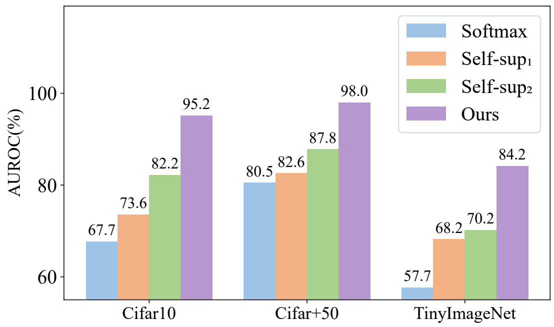


Figure 5: Performance comparisons between our OSSL and two self-supervised learning methods.

Figure 6: Performance comparisons between frozen or non-frozen feature extractor in our OSSL on different benchmark datasets.

tition of the test set. We have two threshold parameters, i.e., γ and μ. Limited by the space, we here take the experiments on Cifar10 and TinyImageNet as the examples. In specific, For the parameter γ, we change its value in the range of $\{0.01, 0.02, 0.03, 0.04, 0.05\}$ both for Cifar10 and TinyImageNet. For the parameter μ, we change its value in the range of $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ for Cifar10 while $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for TinyImageNet. Figure 4 shows the results.

As shown in Figure 4, our model is relatively insensitive to the choice of γ and μ whether on Cifar10 or TinyImageNet. On one hand, it indicates that the selected closed-set classifier in our OSSL has already possessed a significant level of inference capability to some extent. On the other hand, it also provides evidence that we can select an effective threshold without the need for excessive parameter searching. Still, careful selection of these parameters would result in better performance.

4.6 Comparison with Self-Supervised Learning

To further demonstrate the advantages of our learning framework, we here also compare our OSSL with the self-supervised learning method which can use the unlabeled test data as well. In specific, two self-supervised learning methods for utilizing unlabeled test data are provided. One adopts a joint training way (Self-sup$_1$), where the cross-entropy (CE) loss is applied to the labeled training set, and the labeled training set is combined with the unlabeled test set to form a unified unlabeled dataset that is fed into the self-supervised loss (the contrastive loss here). Then these two losses work together to guide the learning of the network. The other one employs a two-stage training way (Self-sup$_2$). In the first stage, the unlabeled dataset composed of the training and test sets is fed into the self-supervised loss to train the network. In the second stage, building upon the model trained in the first stage, the entire network is fine-tuned using the labeled training set with the cross-entropy loss. For the sake of fairness, all these methods, like ours, employ the network architecture in [Vaze *et al.*, 2022]. Limited by the space, we here just provide the results in Cifar10, Cifar+50, TinyImageNet, while for the results on the remaining datasets, please refer to the supplementary materials.

As shown in Figure 5, though the performances of these two self-supervised methods have significantly improved compared to the baseline method, i.e., Softmax, on all benchmark datasets, there is still a significant gap between their performance and our OSSL. We believe that this is mainly attributed to the following reasons: i) There may be some conflicts between CE loss and self-supervised loss during training, thus interfering the learning of the network; ii) The self-supervised proxy tasks may mismatch with the downstream classification task to some extent; iii) The learned features may be too generalized, which is not conducive to the detection of unknown classes.

4.7 Experiment on Frozen Feature Extractor

Despite significant performance breakthroughs achieved by our OSSL, it is worth noting that its training process still requires fine-tuning of the feature extraction network $F(\cdot)$, which can significantly decrease the model's efficiency. One direct way to alleviate this issue is to freeze $F(\cdot)$. To evaluate the performance of OSSL in this situation, we conduct the related experiments and Figure 6 reports the results. We can find that the OSSL with frozen $F(\cdot)$ achieves comparable performance compared to the unfrozen counterpart on all benchmarks. This indicates that in the pursuit of efficiency but with less stringent performance requirements, the model can be trained by directly freezing the feature extractor well-trained on known class data.

5 Conclusion

In this paper, we rethink the OSR problem from the dynamic against dynamic perspective, where an open-set self-learning framework is proposed to construct the dynamic and changeable decision boundaries to adapt changing data distributions rather than a static and fixed ones in existing methods. Extensive experiments verify the effectiveness of our framework, refreshing the new performance records on almost all standard and cross-dataset benchmarks.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (62106102, 62076124, 62376126), in part by the Natural Science Foundation of Jiangsu Province (BK20210292), in part by the Hong Kong Scholars Program under Grant XJ2023035, in part by the RGC-GRF project RGC/HKBU12200820.

Contribution Statement

Haifeng Yang and Chuanxing Geng are Equal Contributions. Songcan Chen is the Corresponding Author.

References

[Bartler *et al.*, 2022] Alexander Bartler, Andre Bühler, Felix Wiewel, Mario Döbler, and Bin Yang. Mt3: Meta test-time training for self-supervised test-time adaption. In *International Conference on Artificial Intelligence and Statistics*, pages 3080–3090. PMLR, 2022.

[Bendale and Boult, 2015] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572, 2015.

[Cao *et al.*, 2022] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *International Conference on Learning Representations*, 2022.

[Cevikalp *et al.*, 2023] Hakan Cevikalp, Bedirhan Uzun, Yusuf Salk, Hasan Saribas, and Okan Köpüklü. From anomaly detection to open set recognition: Bridging the gap. *Pattern Recognition*, 138:109385, 2023.

[Chen *et al.*, 2020] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. *ArXiv*, abs/2011.00178, 2020.

[Chen *et al.*, 2021] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:8065–8081, 2021.

[Deng *et al.*, 2022] Shule Deng, Jin-Gang Yu, Zihao Wu, Hongxia Gao, Yansheng Li, and Yang Yang. Learning relative feature displacement for few-shot open-set recognition. *IEEE Transactions on Multimedia*, 2022.

[Fang *et al.*, 2021] Zhen Fang, Jie Lu, Anjin Liu, Feng Liu, and Guangquan Zhang. Learning bounds for open-set learning. In *International conference on machine learning*, pages 3122–3132. PMLR, 2021.

[Garg *et al.*, 2023] Saurabh Garg, Amrith Rajagopal Setlur, Zachary Chase Lipton, Sivaraman Balakrishnan, Virginia Smith, and Aditi Raghunathan. Complementary benefits of contrastive learning and self-training under distribution shift. *ArXiv*, abs/2312.03318, 2023.

[Geng and Chen, 2022] Chuanxing Geng and Songcan Chen. Collective decision for open set recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):192–204, 2022.

[Geng *et al.*, 2020] Chuanxing Geng, Lue Tao, and Songcan Chen. Guided cnn for generalized zero-shot and open-set recognition using visual and semantic prototypes. *Pattern Recognition*, 102:107263, 2020.

[Geng *et al.*, 2021] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3614–3631, 2021.

[Guo *et al.*, 2021] Yunrui Guo, Guglielmo Camporese, Wenjing Yang, Alessandro Sperduti, and Lamberto Ballan. Conditional variational capsule network for open set recognition. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 103–111, 2021.

[Huang *et al.*, 2022] Hongzhi Huang, Yu Wang, Qinghua Hu, and Ming-Ming Cheng. Class-specific semantic reconstruction for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4214–4228, 2022.

[Iwasawa and Matsuo, 2021] 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.

[Karisani, 2023] Payam Karisani. Neural networks against (and for) self-training: Classification with small labeled and large unlabeled sets. In *Annual Meeting of the Association for Computational Linguistics*, 2023.

[Kong and Ramanan, 2021] Shu Kong and Deva Ramanan. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2021.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[Lake *et al.*, 2015] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332 – 1338, 2015.

[Lee, 2013] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2013.

[Liu *et al.*, 2021] 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.

[Lu *et al.*, 2022] Jing Lu, Yunxu Xu, Hao Li, Zhanzhan Cheng, and Yi Niu. Pmal: Open set recognition via robust prototype mining. In *AAAI Conference on Artificial Intelligence*, 2022.

[Moon *et al.*, 2022] WonJun Moon, Junho Park, Hyun Seok Seong, Cheol-Ho Cho, and Jae-Pil Heo. Difficulty-aware simulator for open set recognition. In *European Conference on Computer Vision*, 2022.

[Mukherjee and Awadallah, 2020] Subhabrata Mukherjee and Ahmed Hassan Awadallah. Uncertainty-aware self-training for text classification with few labels. *arXiv preprint arXiv:2006.15315*, 2020.

[Neal *et al.*, 2018] Lawrence Neal, Matthew Lyle Olson, Xiaoli Z. Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *European Conference on Computer Vision*, 2018.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[Niu *et al.*, 2022] 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.

[Oza and Patel, 2019] Poojan Oza and Vishal M. Patel. C2ae: Class conditioned auto-encoder for open-set recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2302–2311, 2019.

[Perera and Patel, 2021] Pramuditha Perera and Vishal M. Patel. Geometric transformation-based network ensemble for open-set recognition. *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2021.

[Russakovsky *et al.*, 2014] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211 – 252, 2014.

[Scheirer *et al.*, 2013] Walter J. Scheirer, Anderson Rocha, Archana Sapkota, and Terrance E. Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1757–1772, 2013.

[Schneider *et al.*, 2020] 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.

[Scudder, 1965] H. J. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory*, 11:363–371, 1965.

[Sun *et al.*, 2020] 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.

[Vaze *et al.*, 2022] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *the International Conference on Learning Representations*, abs/2110.06207, 2022.

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

[Wang *et al.*, 2022] Zitai Wang, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. Openauc: Towards auc-oriented open-set recognition. *Advances in Neural Information Processing Systems*, 35:25033–25045, 2022.

[Wei *et al.*, 2021] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021.

[Xie *et al.*, 2020] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.

[Xu *et al.*, 2023] Baile Xu, Furao Shen, and Jian Zhao. Contrastive open set recognition. In *AAAI Conference on Artificial Intelligence*, 2023.

[Yoshihashi *et al.*, 2018] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4011–4020, 2018.

[You *et al.*, 2019] Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Universal domain adaptation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2715–2724, 2019.

[Yu *et al.*, 2015] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv*, abs/1506.03365, 2015.

[Zhang *et al.*, 2020] Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *European Conference on Computer Vision*, 2020.

[Zhang *et al.*, 2022] 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.

[Zhou *et al.*, 2021] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2021.

[Zhou, 2022] Zhi-Hua Zhou. Open-environment machine learning. *National Science Review*, 9(8):nwac123, 2022.

[Zhuang *et al.*, 2024] Jingyu Zhuang, Kuo Wang, Liang Lin, and Guanbin Li. Credible teacher for semi-supervised object detection in open scene. 2024.

[Zou *et al.*, 2019] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5982–5991, 2019.