Complex & Intelligent Systems (2025) 11:250
https://doi.org/10.1007/s40747-025-01857-1

ORIGINAL ARTICLE

Energy-based open set domain adaptation with dynamic weighted synergistic mechanism

Zihao Fu[1,2] · Dong Liu[1,3,4] · Shengsheng Wang[2] · Hao Chai[1,2]

Received: 21 September 2024 / Accepted: 1 February 2025 / Published online: 19 April 2025

Abstract

Open Set Domain Adaptation (OSDA) aims to minimize domain variation while distinguishing between known and unknown samples. However, existing OSDA methods, which rely on deep neural network classifiers, often lead to overconfident predictions and fail to clearly demarcate known from unknown samples. To address this limitation, we propose the Energy-based Open Set domain adaptation (EOS) method. EOS introduces a novel two-stage approach involving a separation stage followed by an alignment stage. In the separation stage, we employ an energy-based anomaly detection strategy to identify unknown samples, transforming the traditional K-class classification task into a K+1-dimensional classifier by introducing an additional dimension to model the uncertainty of out-of-distribution samples. To further refine separation, we apply a coarse-to-fine method that iteratively improves the separation outcomes, which are integrated as weighted inputs in the alignment process to enhance feature distribution alignment. In the alignment stage, we employ a dynamic weighted synergistic mechanism, where the separation network and alignment network co-evolve through continuous alternating training. This mechanism enables the system to better adapt to invariant features across domains. We evaluate EOS on standard benchmarks, including Office-31, Office-Home, and VisDA-2017, with experimental results demonstrating that EOS consistently outperforms other state-of-the-art methods.

Keywords Domain adaptation · Open set domain adaptation · Out-of-distribution detection · Energy-based models

Introduction

The widespread application of Domain Adaptation (DA) demonstrates its effectiveness in adapting a model trained on the source domain to perform classification tasks in the target domain [1–3]. Traditional Closed Set Domain Adaptation (CSDA) assumes that the source and target domains share the same category space. However, in practice, the composition of the target domain is often uncertain and may contain Out-Of-Distribution (OOD) samples. As a result, maintaining this assumption becomes challenging. In such cases, applying domain adaptation methods under the closed set condition not only fails to accurately classify the samples but also increases the risk of negative transfer.

To address this issue, Open Set Domain Adaptation (OSDA), first introduced by Busto and Gall [4], provides a promising solution. A commonly used OSDA setting is shown in Fig. 1. In this setting, the target domain contains not only the information present in the source domain but also additional unknown data that does not belong to the source domain's class space. OSDA allows target samples that differ

✉ Dong Liu
liudong@xnu.edu.cn

Zihao Fu
fuzh20@mails.jlu.edu.cn

Shengsheng Wang
wss@jlu.edu.cn

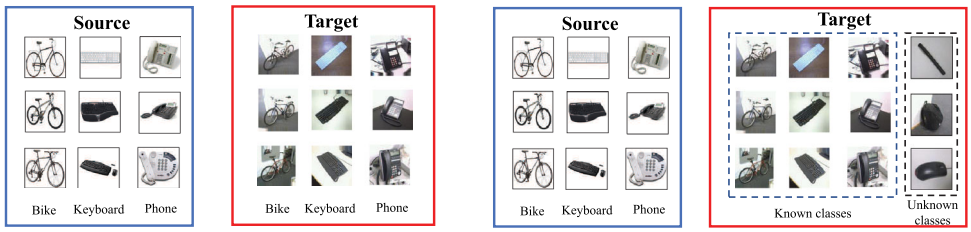
Hao Chai
chaihao21@mails.jlu.edu.cn

1 Hunan Engineering Research Center of Advanced Embedded Computing and Intelligent Medical Systems, Xiangnan University, Chenzhou 423300, China

2 College of Computer Science and Technology, Jilin University, Changchun 130012, Jilin, China

3 School of Computer and Artificial Intelligence, Xiangnan University, Chenzhou 423300, China

4 Key Laboratory of Medical Imaging and Artificial Intelligence of Hunan Province, Xiangnan University, Chenzhou 423300, China



(a) CSDA Setting (b) OSDA Setting

Fig. 1 The figure shows the comparison of CSDA and OSDA. **a** Is the case of Closed Set Domain Adaptation: the source and target domains share the same category space. **b** Is the case of Open Set Domain Adaptation: the target domain contains the same categories as the source domain, in addition to some unknown samples that do not belong to any of the classes in the source domain

from the source domain's label space to be classified into an 'unknown' class, effectively adding an extra dimension to the model to reject identified OOD samples. However, existing methods face a significant challenge: classifiers trained by deep learning networks often become overconfident in their predictions, leading to the misclassification of OOD samples as known classes. This, in turn, exacerbates negative transfer during the alignment process in the second stage.

In recent years, with the success of energy-based anomaly detection methods, we propose using the high or low values of the energy function in the separation stage of OSDA to distinguish whether a sample is in-distribution or out-of-distribution (OOD). The primary cause of overconfidence in existing models is the closed-world assumption inherent in the softmax layer used for multi-class classification tasks, which forces OOD samples in open environments to be classified into known categories. To address this issue, in this thesis, we modify the original K-way classifier into a K+1-dimensional classifier during the separation stage of unknown samples. Instead of relying solely on softmax output, the classification result is determined by the energy value: a higher energy value indicates that the sample is more likely to be OOD, while a lower value suggests that the sample is closer to a known class. This approach effectively models the uncertainty of unknown samples through the newly added dimension.

In this paper, we propose a two-stage approach called EOS, which leverages an energy-based anomaly detection method to address OSDA. EOS is an end-to-end network module, where the first stage trains a separation network, and the second stage trains an alignment network while fine-tuning the separation module. In the first stage, to improve the classifier's ability to identify unknown samples, an additional dimension is introduced to model the uncertainty in an open-world environment. The effective sample size in this extra dimension is increased by adding noise to the source domain. This leads to a classifier capable of effectively distinguishing unknown samples, although it may only coarsely classify some unknown samples due to distribution differences between the source and target domains. To enhance precision, we incorporate a more refined separation mechanism by ranking samples based on energy function values derived from the logit layer. Since the lowest and highest energy values correspond to known and unknown class samples, respectively, this ensures that unknown samples are excluded during the second stage of domain alignment. In the alignment stage, we employ a dynamic weighted synergistic mechanism, where the separation network and alignment network co-evolve through continuous alternating training, allowing the system to better adapt to changing feature distributions. We construct an adversarial generative network for the separation stage. Through continuous alternating training, we obtain a classifier that can accurately determine whether a target sample belongs to the known label space. Samples that do not belong to a known class are classified as unknown and rejected. we continue optimizing the separator during the alignment stage to better adapt to the evolving generative adversarial network.

This paper makes the following key contributions:

(1) We propose a novel two-stage OSDA framework, comprising a separation stage and an alignment stage. In the separation stage, we introduce an innovative energy-based anomaly detection strategy, which extends the traditional classification task by incorporating an additional dimension specifically designed to capture the uncertainty of OOD samples.
(2) During the alignment stage, we leverage a dynamic weighted synergistic mechanism, where the separation network is continuously co-evolved and refined alongside the alignment network, allowing the energy-based classifier to flexibly adapt to shifting feature distributions across domains.
(3) We conducted extensive experiments on several standard benchmarks, including Office-31, Office-Home, and VisDA-2017. The results show that our method consistently outperforms other advanced approaches,

demonstrating its strong ability to balance classification accuracy and OSDA.

The following structure of this paper is set up as follows: Sect. "Relate work" provides an overview of the preceding relevant research. Section "Method" delves into the details of our proposed methodology. Section Experiment presents comprehensive experimental results followed by in-depth analyses. Section "Conclusion" concludes with a summary of our research and outlines directions for future research.

Relate work

Closed set domain adaptation (CSDA)

In the CSDA setting, the source and target domains share the same label space. All samples in the source domain are labeled, while the target domain has no annotations. Existing methods for CSDA can be broadly categorized into three types:

Discrepancy-based Methods: These methods aim to minimize the difference between feature distributions in the source and target domains. For instance, the unsupervised domain adaptation method DDC [5] utilizes Maximum Mean Discrepancy (MMD) [6], which maps data from both domains into a reproducing kernel Hilbert space and computes the difference between their mean feature representations. By reducing this distance, domain shift is minimized. Additionally, TPDS [7] introduces a Target Prediction Distribution Searching paradigm for Source-Free Domain Adaptation. It uses proxy distributions and pairwise alignment with category consistency to minimize adaptation errors, achieving state-of-the-art performance.

Adversarial-based Methods: Inspired by Generative Adversarial Networks (GANs) [8], these methods learn domain-invariant features through adversarial training. A notable example is DANN [9], which incorporates a domain discriminator sub-network to distinguish the data source, while a generative network is trained to confuse the discriminator. SPA [10] introduces a graph spectral alignment framework that balances transferability and discriminability by aligning domain graphs in eigenspaces and refining the target domain structure with fine-grained message propagation. S2ADAP [11] further extends this approach with a privacy-preserving, source-free domain adaptation framework for person re-identification (ReID). It uses GAN-based style diversity augmentation and adversarial mutual teaching to address inter- and intra-domain style discrepancies, enhancing target adaptation without requiring source data.

Reconstruction-based Methods: These methods focus on improving pseudo-labeling and distribution alignment through reconstruction. CPD [12], for example, introduces a class prototype discovery method for source-free domain adaptation. By optimizing semantic class prototypes, it addresses limitations in existing self-training and data generation methods.

However, the above methods are not suitable for Open Set Domain Adaptation (OSDA) due to the presence of unknown samples. These unknowns increase negative transfer, making it challenging to directly extract domain-invariant features.

Open set domain adaptation (OSDA)

The target domain of OSDA is set closer to the real-world situation. The target domain contains OOD samples outside the sample space of the source domain, this constraint was first proposed by Saito et al. [13] and is also used in our paper. ATI-λ [4] constructs a function from the source domain to the target domain and optimizes the distance between the two domains by continuously training. the distance can determine whether this sample belongs to the unknown class or the source class. OSBP [13] cleverly combines the idea of adversarial, by adding an extra dimension to the classifier to represent the probability that a sample belongs to an unknown class. The method chooses a threshold value to indicate whether the current sample should be identified as unknown or not, and the choice of the threshold value largely determines the accuracy of the classification and also indicates the limitations of the method. STA [14] avoids the problem of hyperparameter selection. It computes the distance between each data and source class through a stepwise fine separation process, selects the closest and farthest sample set in the target domain as known and unknown data, and uses that distance as an instance-level weight in the adversarial network to achieve alignment of the two domains. ROS [15] exploits the inherent properties of self-supervision to solve the OSDA problem. Not only a more robust OSDA metric (HOS) is defined, but also the rotation classification method in self-supervised training is used to distinguish OOD samples. MAOSDAN [16] introduces attention-aware backpropagation for distinguishing unknown classes, auxiliary adversarial learning to mitigate negative transfer, and adaptive entropy suppression to enhance classification confidence and accuracy in complex adaptation tasks.However, the above method leads to a model that is always overconfident in its predictions due to softmax classifiers forcing classification into a known class when separating unknown samples.

Energy-based models (EBMs)

EBMs capture the dependencies between variables by imposing a range-constrained energy on each configuration of the variables, which has been implemented extensively in the field of generative modeling [17–19]. In [20], the authors

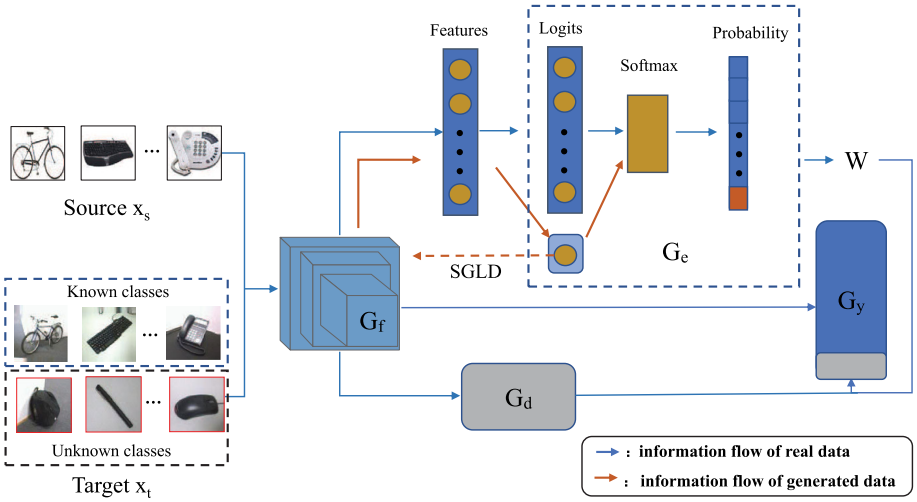


Fig. 2 The overall framework diagram for EOS. G_e in the figure is our energy-based separation module. The data first enters our feature extractor G_f and then flows into G_e, which is combined with an energy model to improve the recognition of unknown classes, resulting in a weight w. The generative adversarial network G_d and the classifier G_y are the network models of our alignment stage. w is used in this stage as instance-level weights to participate in the process of domain adaptation

theoretically demonstrate that OOD data can easily lead to over-prediction by softmax classifiers. JEM [21] successfully calibrated the classification accuracy of the model by optimizing the joint distribution with an energy-based function [22]. Later, EOW-softmax [23] replaced softmax classifiers in the closed world by combining the results generated by the logits layer with the energy function, solving the problem of high confidence and also better detecting OOD samples. In this paper, we also use an energy-based loss function. When putting an unknown sample into an energy model trained in the source domain, the higher energy values are more likely to be unknown samples, and the lower energy values are more likely to be known samples. Since it is very difficult to get OOD samples that can participate in the training directly in the energy model [24], we use a sampler called SGLD [25] to generate samples. This method, during training, injects noise into the parameter updates and anneals the step size.

Method

In this section, we first introduce the basic settings, provide an overview of EOS, and present an energy separation strategy. Then, we offer the dynamic weighted synergistic mechanism in detail. Finally, we reveal the complete two-stage training process. The framework of EOS is shown in Fig. 2.

Preliminaries

We define a source domain dataset $\mathcal{D}_s = \{(x_s, y_s) : x_s \sim p_x, y_s \sim p_{y|x}\}$ that has been tagged and a target domain dataset $\mathcal{D}_t = \{x_t : x_t \sim p_t\}$ that does not contain labels at all. x_s denotes the sample in the source domain, while y_s denotes the label of the corresponding sample in the source domain. The data in the target domain we denote by x_t and these samples are without labels. p_s and p_t show the distribution of edge inputs in the source and target domains, respectively, where we explicitly specify $p_s \neq p_t$. We represent the conditional output distribution of the source domain as $p_{y|x}$. However, we cannot provide the conditional output distribution for the target domain because we lack the true labels needed to calculate it. In CSDA, for the label set $\mathcal{C}_s$ of the source domain and the label set $\mathcal{C}_t$ of the target domain, we then obtain the relationship: $\mathcal{C}_s = C_t$. However, due to the presence of unknown samples, in OSDA, it directly results that the class space of the target domain is contained in the source domain, i.e., $\mathcal{C}_s \subset C_t$. We treat the classes in $\mathcal{C}_s$ as known classes and those belonging to $\mathcal{C}_s \setminus C_t$ as unknown classes. Also due to unknown samples, we can know that $p_s = p_t^{\mathcal{C}_s}$ which is denoted as the edge input distribution of known category samples in the target domain. This means that the main task of our OSDA is not only to achieve the alignment of known categories but also to reject unknown samples from participating in this process.

Overview

The two major challenges of OSDA are the domain discrepancy problem and the inability to effectively discriminate between known and unknown classes, respectively. When matching the whole source and target domains directly ignoring the unknown samples, this mandatory alignment method directly leads to the unknown samples directly participating in the training process of the known samples, which leads to negative migration. Thus, we should only focus on the shared category space of the two domains in the alignment process and reject the unknown samples. That is, if we can find an effective separation strategy to separate the unknown samples first, then we can use the traditional SCDA-related methods in the alignment process afterward. Based on the above analysis, we decide to use a two-stage approach (EOS). EOS uses an energy-based OOD detection method in the separation

stage, and two progressively finer separations to first train a classifier that can effectively identify unknown samples. Then, we use the known target domain samples predicted in the first stage to align with the source domain, while training an extra dimension in the network to model the unknown samples, and finally, reduce the domain differences by adversarial learning to also reject the unknown samples.

Energy separation strategy

As mentioned above, traditional softmax classifiers force an unknown sample to be classified into one of the known classes due to the uncertainty of the open world. We used a new energy-based softmax classifier [23] that successfully identified the unknown samples. The network model changes the traditional softmax layer by adding an extra dimension to the logits layer to model the unknown samples, as shown in Fig. 2, where the traditional K-way classification is changed to K+1 dimensions and the final softmax layer produces probabilistic results for K+1 classes. We first assume that our network model is f: $\mathbb{R}^D \rightarrow \mathbb{R}^{K+1}$. We stipulate that this model is an incomplete model that does not include the final softmax first, so the model yields only K+1 logits layer processed values. For any one sample x, $f(x)[i]$ denotes the corresponding i-th logit, with $i \in \{1, ..., K, K+1\}$. Finally, our complete model $F_\theta(x)$ is composed of a combination of $f(x)$ and a softmax layer. The results obtained are shown below:

$$F_\theta(x)[i] = \frac{exp\,(f(x)[i])}{\sum_{j=1}^{K+1} exp\,(f(x)[j])} \tag{1}$$

With the help of EBMs [22] and EOW-softmax [23], we obtain an energy function formula:

$$E_\theta(x) = \log F_\theta(x)[K+1] \tag{2}$$

and the final energy-based optimization function Le:

$$\begin{aligned} \min_\theta \mathbb{E}_{p(x)} \Big[-\log F_\theta(x)[y] \Big] \\ + \mathbb{E}_{p_{\bar{\theta}}(x)} \Big[-\log F_\theta(x)[K+1] \Big] \end{aligned} \tag{3}$$

For equation (2), [23] explains that $E_\theta(x)$ can assign high energy to variables that are not observed (i.e., unknown samples), and conversely lower$E_\theta(x)$ is more likely to be a sample of a known category. For Equation (3), the first term optimizes the classification accuracy of the first K classes by using the true labels of the source domain data to calculate the maximum log-likelihood objective, and the second term indicates that we optimize the probability prediction for the K+1-th term, which belongs to the unknown class. However, we notice that there are no directly labeled unknown samples for our model to use, so we take a sampler based on SGLD to generate unknown samples. The network parameters of the second term are the same as those of the first term, but the parameters of the second term are frozen because it is necessary to ensure that the noise samples are generated on the basis of the samples from the source domain and cannot participate in the gradient update to avoid negative effects on the optimization of the first K classes. It is also explained theoretically in EOW-softmax that optimizing the final result of our formulation (4) can make the softmax score of the first K categories proportional to our edge input distribution p_s, while the result of the K+1-th category is inversely proportional to p_s.

At this point, according to the energy value we can find out which data are closer to the shared class and which are more likely to be unknown samples, but for a significant portion of the data, we cannot determine their classification when their probability of belonging to the K+1-th class is close to 0.5, so we propose a finer second separation. In the final obtained softmax scores, based on the value of the K + 1-th class sorted from largest to smallest, we choose only the highest and lowest sorted data, which have high confidence levels, at which point we can basically determine that these data belong to the unknown and known samples, respectively. These samples are then pseudo-labeled and used as input for the second stage of alignment training.

Dynamic weighted synergistic mechanism

After the process of separation network G_e in the first stage, the scene is now more like a traditional CSDA. Dynamic Weighted Synergistic Mechanism (DWSM) in doing domain alignment uses adversarial domain adaptation [9, 13, 14], where we use a feature extractor G_f to obtain domain invariant features and a classifier G_y to calculate the final classification accuracy. Since the source domain has $|\ C_s\ |$ categories, the final classification result is of $|\ C_s\ |$+1 dimensions, with extra dimensions to calculate the likelihood of belonging to unknown samples. We first define the classification loss on the source domain as follows:

$$L_{cls}^s = \frac{1}{n_s} \sum_{x_i \in \mathcal{D}_s} L_y \left(G_y^{1:|C_s|} \left(G_f\,(x_i) \right), y_i \right) \tag{4}$$

where L_y is cross-entropy loss, also, since the source domain does not contain unknown samples, we use $G_y^{1:|C_s|}$ to denote the probability that a sample belongs to each of the first $|\ C_s\ |$ known classes. Based on our separation results, the probability of a sample belonging to an unknown class that can be clearly known is expressed using w, i.e., $w_i = G_y^{1:|C_s|} \left(G_f\,(x_i) \right)$, which represents the probability that the energy-based separator G_e assigns a sample to an unknown class. The higher the value of w, the higher the probability

of belonging to the unknown class. Similarly, 1-w implies the probability that this sample belongs to the known class. In this way, when doing a multi-class classification tasks, we don't need to train another binary classifier to indicate Whether the current data is an OOD sample. To more accurately use the data from the target domain in the calculation of the adversarial loss, we use its corresponding w as a soft instance-level weight, as follows:

$$L_d = \frac{1}{n_s}\sum_{x_i \in \mathcal{D}_s} L_e\left(G_d\left(G_f\left(x_i\right)\right), d_i\right) + \frac{1}{\sum_{x_j \in \mathcal{D}_t}\left(1-w_j\right)}\sum_{x_j \in \mathcal{D}_t} L_e\left(G_d\left(G_f\left(x_i\right)\right), d_j\right) \tag{5}$$

where G_d denotes our domain discriminator network, G_f is our feature discriminator, and d_i is used to indicate whether the current data is from the source or target domain, corresponding to a value of 0 and 1, respectively.

In addition, we need to train the unknown samples that can be effectively distinguished by the classifier G_f. L_y ensures that the model is very accurate in classifying shared classes, but is unable to model the dimensionality corresponding to unknown samples. After we sort the samples in the separation stage based on w values from largest to smallest, the top few samples belong to the unknown class with a very high confidence level. Using these data and assigning w to each data as a weight, we can define a weighted loss used to distinguish the unknown class:

$$L_{cls}^t = \frac{1}{\sum_{x_j \in \mathcal{D}_t}\left(1-w_j\right)}\sum_{x_j \in \mathcal{D}_t}\left(1-w_j\right) H\left(G_y^{1:|C_s|}\left(G_f\left(x_j\right)\right), y_u\right) \tag{6}$$

where y_u denotes the unknown class and this loss assigns samples with high w values to the unknown class. $G_y^{|C_s|+1}\left(G_f\right)$ is the probability of assigning the target sample to the unknown class by the classifier G_y. Finally, we define an entropy loss that forces the decision boundary to pass through the low density region of the target domain[26, 27].

$$L_h = \frac{1}{\sum_{x_j \in \mathcal{D}_t}\left(1-w_j\right)}\sum_{x_j \in \mathcal{D}_t}\left(1-w_j\right) H\left(G_y^{1:|C_s|}\left(G_f\left(x_j\right)\right)\right) \tag{7}$$

where H is the entropy loss and $H_{(p)} = -\sum_k p_k \log p_k$. In optimizing this loss, we only need to minimize the entropy of the data belonging to known classes in the target domain, so we use $1-w$ as the instance-level weights of the samples. These are the loss functions we need to use in the domain alignment stage, and minimizing these losses can achieve our goal of multi-class classification tasks under OSDA. However, there is still a problem in our experiments, which directly leads to the deterioration of our classification accuracy. In order to get better performance and faster convergence of the classifier G_y in the source domain during the alignment stage, the feature extractor G_f in this stage directly uses the network model and parameters from the completion of the separation stage. We keep obtaining invariant features in both domains through the adversarial network during the domain adaptation process. After many iterations, G_y obtains better predictions using the features extracted by G_f, but this also directly leads to a significant change in the distribution of the final extracted features compared to the beginning of the second stage, as shown in Fig. 3. This means that if G_e still uses G_f to predict the energy separation values, the confidence level will be reduced. To address this issue, we alternate training G_e once after each round of training the ground adversarial network in the second stage to ensure that the accuracy of our ground energy separation prediction does not decrease due to changes in G_f. When we need to update the network parameters of G_e, the parameters of G_f are frozen because we want G_e to achieve better separation using domain invariant features instead of only performing well on the source domain, thus avoiding negative migration while achieving domain alignment.

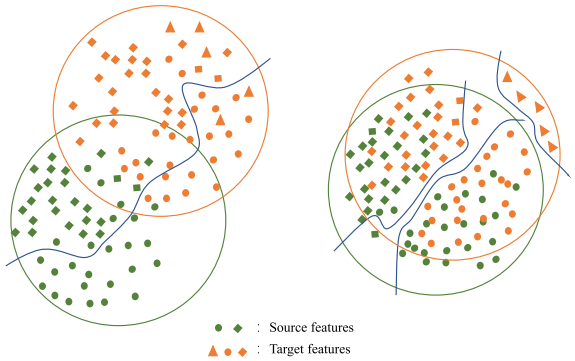


Fig. 3 The diagram of the second-stage learning process. Since the domain-invariant features are learned through the adversarial network, which are different from the features trained solely on the original data, we further adapt the model to these domain-invariant features in the second stage by freezing the parameters of G_f and retraining the energy-based separation module G_e

Training process

Step 1

The main function of this stage is to be able to separate the known samples from the unknown samples. We use the energy-based softmax function to finally determine a feature extractor G_f and a classifier G_e. Our goal in this stage is

to minimize the classification loss L_e of G_e to improve the accuracy of separating unknown samples as much as possible. If we use θ_f, θ_e to denote the optimal parameters of G_f and G_e, respectively, then our objective can be expressed by the following equation:

$$\left(\hat{\theta_f}, \hat{\theta_e}\ |_{c=1}^{|C_s|}\right) = \underset{\theta_f, \theta_e|_{c=1}^{|C_s|}}{\arg\min}\ L_e \tag{8}$$

Step 2

This stage starts with training a classifier G_y that performs well on the source domain based on G_f in the first stage. We then obtain the weights w we need by processing the extra dimensions of G_e in the first stage, which will be added to L_d and L_h to be used as instance-level weights. Then, in the adversarial, we get the domain invariant features by optimizing L_d, while we have to continue minimizing Le again when G_f keeps changing because of L_d. These two processes are performed alternately. If $\theta_f, \theta_y, \theta_d$ are used to denote the most available parameter values of G_f, G_y, and G_d respectively, then our final optimization objective can be written as:

$$\left(\hat{\theta_y}, \hat{\theta_d}\right) = \underset{\theta_y, \theta_d}{\arg\min}\ L_{cls}^s + L_{cls}^t + \lambda L_d + L_h \tag{9}$$

$$\left(\hat{\theta_f}\right) = \underset{\theta_f}{\arg\min}\ L_{cls}^s + L_{cls}^t - \lambda L_d + L_h \tag{10}$$

$$\left(\hat{\theta_e}\right) = \underset{\theta_e}{\arg\min}\ L_e \tag{11}$$

where λ is a hyperparameter used to measure our adversarial loss.

We propose a two-stage training model. In the first stage, we introduce an energy-based separation strategy to develop a classifier that effectively distinguishes between known and unknown samples. In the second stage, we refine the separation results from the first stage by leveraging DWSM, which simultaneously improve the performance of the unsupervised multi-class classification task.

Theoretical complexity analysis

First stage: energy function and pseudo-label generation

- **Energy function calculation**: Analyze the time complexity of computing the energy value for each sample through the classifier. Suppose that the forward propagation of the classifier has a complexity of $O(f(n))$, where n is the dimensionality of the input features, the complexity for m samples would be $O(m \cdot f(n))$.
- **Pseudo-label generation**: In secondary separation, the sorting based on energy values may have a complexity of $O(m \log m)$.

Second stage: adversarial network training

- **Adversarial network complexity**: Adversarial training involves alternating optimization of the generator and discriminator. For each training batch, analyze the forward and backward propagation complexity. Assuming the model has k parameters, the complexity for a batch size b is $O(k \cdot b)$.
- **Dynamic weight updates**: The complexity of computing instance-level weights. Calculating weights based on alignment loss have a complexity of $O(m)$.

Overall complexity Summing up the components, the overall complexity of the algorithm can be expressed as:

$$O_{\text{EOS}} = O(m \cdot f(n)) + O(m \log m) + T \cdot (O(k \cdot b) + O(m)) \tag{12}$$

where T is the number of iterations in adversarial training.

Experiment

Experimental details

We applied our method on three widely used standard data sets Office31, Office-Home, and VisDA-2017, and compared it with other state-of-the-art methods. Office31 [28] contains three different domains, Webcam (W), Amazon (A), and Dslr (D), and each domain contains 31 categories, and all three domains contain the same category space. In our experiments, we use the same setting as OSBP. We sorted all classes by initial alphabet and selected the first ten classes at the top as known classes and the last eleven classes as unknown classes. We finally completed a multiclassification task with 11 categories in the Office31 dataset. Office-Home [29] contains 4 different domains, namely Product (Pr), Art (Ar), Real World (Rw), and Clipart (Cl), and each domain contains 65 categories. Our method selects the first 25 categories in alphabetical order as known classes and all other category samples as unknown classes. The number of classes and domain differences on this dataset exceeded those on the Office31 dataset. VisDA-2017 [30] contains two domains, Synthetic and Real, each containing 12 classes. The Synthetic one is a series of 2D renderings synthesized from 3D items, and the Real one is composed of real images. We run only one task on this dataset, where the source domain uses Synthetic and the target domain uses Real. For the division

of known and unknown classes, we use the same configuration as in STA. For the known classes we choose bicycle, bus, car, motorcycle, train, and truck, and the remaining six classes are considered as unknown classes. Compared to the two datasets of the Office series, the experiments on VisDA-2017 are more challenging due to the large domain differences between the synthetic and real data. The three datasets have different degrees of distributional differences, which also ensures that our approach can be evaluated more comprehensively.

EOS is compared to the following methods: OSVM [31], MMD + OSVM, DANN + OSVM,ATI-λ [4], OSBP [13], STA [14], UAN [32], SE-CC [33], DAOD [34], ROS [15], and SURC[35]. We apply the maximum mean discrepancy [6] to OSVM to obtain MMD + OSVM. Similarly, DANN + OSVM is the method obtained by combining the domain adversarial network [9] and OSVM. The backbone network used in our paper on all three datasets is the pretrained Resnet-50 [36]. In our experiments, we choose different random seeds to run three times and finally record the average of the three results.

Building on previous work [13, 15]: we chose three metrics: (1) OS*:averaging the accuracy over all known classes; (2) UNK: the accuracy of the network over unknown classes. (3) HOS: we use this metric to estimate the normalized accuracy over all samples, which is calculated based on OS* and UNK, as follows:

$$HOS = \frac{2 \times OS^* \times UNK}{OS^* + UNK} \tag{13}$$

In our experiments, we record OS*, UNK, and HOS for each domain adaptation task and calculate the average accuracy of all tasks on a dataset. In this study, we focus on HOS evaluation metrics. Our initial learning rate is set to 0.001 and the weight decay of the learning rate is 0.0005. the gradient descent algorithm uses momentum SGD and the momentum is set to 0.9.

Results

The experimental results on Office-31 are shown in Table 1, where we compared all six subtasks. Our method achieved the highest precision on four out of the six tasks, and the average HOS of our method is 2.0% higher than that of the second-best method, ROS. Additionally, it can be observed that methods such as STA, UAN, SE-CC, and SUCR exhibit a significant gap between their OS* and UNK performance, which is primarily due to their poor classification capability for unknown target domain samples. In contrast, when comparing these two metrics for our method, the smaller gap indicates that our method better distinguishes known and unknown samples.

Table 1 Accuracy (%) of all methods on Office-31 dataset with ResNet50 backbone

Office-31	A→D			A→W			D→A			D→W			W→A			W→D			Avg.		
	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS	OS*	UNK	HOS
STAsum [14]	95.4	45.5	61.6	92.1	58.0	71.0	94.1	55.0	69.4	97.1	49.7	65.5	92.1	46.2	60.9	96.6	48.5	64.4	**94.6**	50.5	65.5
STAmax [14]	91.0	63.9	75.0	86.7	67.6	75.9	83.1	65.9	73.2	94.1	55.5	69.8	66.2	68.0	66.1	84.9	67.8	75.2	84.3	64.8	72.5
OSBP [13]	90.5	75.5	82.4	86.8	**79.2**	82.7	76.1	72.3	75.1	97.7	**96.7**	97.2	73.0	74.4	73.7	99.1	84.2	91.1	87.2	80.4	83.7
UAN [32]	**95.6**	24.4	38.9	**95.5**	31.0	46.8	93.5	53.4	68.0	99.8	52.5	68.8	**94.1**	38.8	54.9	81.5	41.4	53.0	93.4	40.3	55.1
SE-CC [33]	84.2	64.4	73.0	84.0	46.6	59.9	96.6	55.9	70.8	99.1	73.8	84.6	90.3	12.2	21.5	85.9	50.7	63.8	90.0	50.6	62.3
SUCR [35]	89.6	69.8	78.5	86.6	63.5	73.3	82.6	44.1	57.5	96.7	94.5	95.6	83.9	69.6	76.1	99.4	99.9	99.6	89.8	74.4	81.4
ROS [15]	87.5	77.8	82.4	88.4	76.7	82.1	74.8	81.2	77.9	99.3	93.0	96.0	69.7	**86.6**	77.2	**100.0**	**99.4**	**99.7**	86.6	85.8	85.9
DAOD [34]	84.2	84.2	84.2	89.8	75.5	82.0	71.8	80.6	75.9	98.0	76.0	85.6	72.9	87.2	**79.4**	97.5	84.3	90.4	85.7	81.3	82.9
Ours	91.9	**88.5**	**90.2**	91.9	77.5	**84.1**	**100.0**	**98.0**	**99.0**	**100.0**	96.6	**98.3**	75.1	79.9	77.4	74.2	83.7	78.6	88.8	**87.3**	**87.9**

Boldface values indicate the best performance among the compared methods

Table 2 Accuracy (%) of all methods on Office-Home dataset with ResNet50 backbone

Office-Home	Pr → Rw			Pr → Cl			Pr → Ar			Ar → Pr			Ar → Rw			Ar → Cl		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
ResNet [36]	70.2	61.0	65.3	32.8	67.1	44.1	45.7	70.5	55.5	64.4	64.0	64.2	76.9	59.7	67.2	44.8	68.7	54.2
DANN [9]	77.4	48.4	59.5	50.5	49.9	50.2	61.6	54.5	57.8	71.0	38.9	50.2	75.0	50.1	60.1	54.6	48.8	51.6
STA [14]	76.2	64.3	69.5	44.2	67.1	53.2	54.2	72.4	61.9	68.0	48.4	54.0	78.6	60.4	68.3	46.0	72.3	55.8
OSBP [13]	76.2	71.7	73.9	44.5	66.3	53.2	59.1	68.1	63.2	71.8	59.8	65.2	79.3	67.5	72.9	50.2	61.1	55.1
UAN [32]	84.0	0.1	0.2	59.1	0.0	0.0	73.7	0.0	0.0	81.1	0.0	0.0	88.2	0.1	0.2	62.4	0.0	0.0
DAOD [34]	84.1	34.7	49.1	60.0	36.6	45.5	66.7	43.3	52.5	72.6	51.8	60.5	78.2	62.6	69.5	55.3	57.9	56.6
ROS [15]	70.8	78.4	**74.4**	46.5	71.2	56.3	57.3	64.3	60.6	68.4	70.3	69.3	75.8	77.2	**76.5**	50.6	74.1	60.1
Ours	71.7	75.6	73.6	47.8	73.6	**57.9**	54.2	76.7	**63.5**	67.5	77.1	**72.0**	70.4	76.6	73.4	59.8	71.8	**65.2**

	Rw → Ar			Rw → Pr			Rw → Cl			Cl → Rw			Cl → Ar			Cl → Pr			Avg.		
	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**	OS*	UNK	**HOS**
ResNet	61.7	63.5	62.5	74.4	58.4	65.4	40.7	54.6	46.7	59.5	68.8	63.8	40.1	76.1	52.5	51.6	67.8	58.6	55.2	65.0	58.3
DANN	67.3	51.9	58.6	80.8	46.6	59.1	59.5	49.3	53.9	73.5	55.2	63.1	57.6	56.7	57.1	66.2	45.0	53.6	66.3	49.6	56.2
STA	67.5	66.7	67.1	77.1	55.4	64.5	49.9	61.1	54.5	67.0	66.7	66.8	51.4	65.0	57.4	61.8	59.1	60.4	61.8	63.3	61.1
OSBP	66.1	67.3	66.7	76.3	68.6	72.3	48.0	63.0	54.5	72.0	69.2	70.6	59.4	70.3	64.3	67.0	62.7	64.7	64.1	66.3	64.7
UAN	77.5	0.1	0.2	85.0	0.1	0.1	66.2	0.0	0.0	80.6	0.1	0.2	70.5	0.0	0.0	74.0	0.1	0.2	75.2	0.0	0.1
DAOD	71.3	50.5	59.1	81.8	50.6	62.5	58.4	42.8	49.4	77.8	57.0	65.8	59.1	61.7	60.4	70.8	52.6	60.4	69.6	50.2	57.6
ROS	67.0	70.8	68.8	72.0	80.0	75.7	51.5	73.0	60.4	65.3	72.2	**68.6**	53.6	65.5	58.9	59.8	71.6	65.2	61.6	72.4	66.2
Ours	59.7	83.8	**69.7**	70.6	79.1	**74.6**	53.4	73.3	**61.8**	62.9	74.8	68.3	53.2	70.7	**60.7**	59.0	76.5	**66.6**	60.8	75.8	**67.3**

Boldface values indicate the best performance among the compared methods

Table 3 Accuracy (%) of all methods on VisDA-2017 with ResNet50 backbone

Synthetic→Real									
	bcycl	Bus	Car	mcycl	Train	Truck	OS*	UNK	HOS
ResNet [36]	42.6	6.4	30.5	67.1	84.0	0.2	34.7	55.1	42.6
DANN [9]	20.1	71.4	29.5	74.4	67.8	10.4	45.6	78.9	57.8
OSVM [31]	31.7	51.6	66.5	70.4	**88.5**	20.8	54.9	38.0	44.9
MMD+OSVM [31]	39.0	50.1	64.2	79.9	86.6	16.3	56.0	44.8	48.3
DANN+OSVM [31]	31.8	56.6	**71.7**	77.4	87.0	22.3	57.8	41.9	48.6
ATI-λ [4]	46.2	57.5	56.9	79.1	81.6	**32.7**	59.0	65.0	61.9
OSBP [13]	53.9	77.6	56.4	**89.1**	74.4	22.2	**62.3**	71.3	66.5
STA [14]	38.2	69.1	51.2	87.6	78.0	11.1	55.9	75.2	64.1
Ours	**61.1**	**75.6**	52.3	70.7	85.3	20.8	61.0	**90.8**	**73.0**

Boldface values indicate the best performance among the compared methods

Table 2 presents the experimental results on Office-Home, where the domain differences are significantly larger. In these experiments, our method achieved the highest average HOS, reaching 67.3%. While ROS achieved the highest average HOS among all baseline methods, it is still 1.0% lower than our method. Our proposed method outperformed all other comparison methods in 8 out of the total 12 tasks. Additionally, it can be observed that methods such as DANN, UAN, and DAPD performed even worse than ResNet-50. This is primarily due to their incorrect alignment of source domain features with unknown target features, which resulted in negative transfer. The failure of these methods can be attributed to their poor ability to separate known and unknown target domain samples. A similar phenomenon to that observed in Table 1 occurs here, where some methods show high accuracy on the OS* metric but perform poorly on the unknown class. For instance, the UNK results of UAN on all 12 tasks are essentially 0, indicating that this method is unable to effectively separate unknown samples. In contrast, the average UNK of our method is the highest, exceeding the second-best method, ROS, by 3.4%. Furthermore, it is observed that the OS* and HOS metrics of our method are nearly identical, demonstrating that our method achieves balanced classification performance for both known and unknown classes.

Table 3 shows the experimental results on VisDA-2017. We present the classification accuracy for all classes, and the results demonstrate that the average HOS of our method is the highest, reaching 73.0%. This further confirms that EOS remains effective on datasets with significant domain discrepancies. Additionally, considering the experimental results from Tables 1, 2, and 3, it can be observed that the main advantage of our method lies in its ability to effectively distinguish between known target domain samples and unknown samples. This is particularly evident in Table 3, where some comparison methods achieve average OS* scores similar to ours, but our method achieves a much higher UNK score. This result indicates that our method is highly effective in separating known and unknown target domain samples.

Analysis

Ablation studies

Table 4 shows the results of the experiments on Office-Home for our method and its related variants. We tested all 12 subtasks on this dataset. In the table, we use HOS metrics to compare the differences. (1) w/o s is a variant that we use the traditional softmax classifier instead of the energy-based classifier we propose in this thesis, and in the second stage we use only the adversarial network, eliminating the process of optimizing the first stage of the separation network. (2) w/o c is a variant that we propose using the energy separation strategy, but the alignment stage does not alternatively train the separation network, only the adversarial network is used. (3) w/o e is a variant we propose based on w/o s, which only adds the optimization training of the first-stage separation network in the alignment stage, and the other settings are the same as w/o s.

Studying the data in Table 4 we find that our proposed method improves by 9.9% relative to w/o s. This is also direct evidence that our proposed two innovations can effectively improve the classification accuracy under the OSDA problem. In addition, comparing w/o s and w/o c we find that the improvement on HOS reaches 6.9%. By comparing these two experiments, we confirm that the traditional softmax network cannot effectively distinguish the unknown samples, while the energy-based separation strategy proposed in this paper can effectively separate the unknown samples. Besides, when comparing w/o s and w/o e, we notice that w/o e is only 1.6% higher than w/o s in terms of average HOS, and this improvement is relatively small compared to the other variants. Because w/o e is only optimized on the traditional softmax, which does not effectively distinguish

Table 4 Results of EOS and its three variants on Office-Home (Resnet-50)

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Ar→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
EOS w/o s	66.3	45.2	56.1	63.9	62.6	55.3	62.5	62.1	47.3	60.8	52.7	54.5	57.4
EOS w/o c	70.8	54.3	59.2	69.6	71.2	63.9	66.9	72.4	57.1	65.2	57.8	63.3	64.3
EOS w/o e	67.4	45.7	57.6	65.8	63.2	57.1	63.8	65.9	49.5	61.7	54.4	56.3	59.0
EOS	**73.6**	**57.9**	**63.5**	**72.0**	**73.4**	**65.2**	**69.8**	**74.6**	**61.8**	**68.3**	**60.7**	**66.6**	**67.3**

Boldface values indicate the best performance among the compared methods

unknown samples, this also leads to the fact that we are optimizing the alignment stage under the influence of negative migration, so the accuracy of the improvement is not high. Finally, we compare w/o s, w/o c, and our method, and the experimental results also prove that both of our innovations combined are better than adding only any one of them.

Parameter sensitivity

In Fig. 4a we show the effect of different values of the hyperparameter λ (in(9)) on the accuracy of our experiments. We randomly selected three tasks on Office-Home, and in this experiment, we choose Pr-Rw, Ar-Pr, and Rw-Cl. The values of λ were 0.5, 1, 2, 4, 6, 8, and 10, respectively, and the changes in experimental results of the three tasks were recorded by HOS indicators. In the figure, we find that the accuracy of the experiment increases and then decreases as λ changes from 0.5 to 10. The experimental accuracy is highest when $\lambda = 2$. The results show that the adversarial network helps us to improve the overall accuracy, and the effect of adversarial on the overall results is nonlinear. Figure 4b shows the changes of three metrics OS*, UNK, and HOS for a subtask Cl-Rw in Office-Home under the influence of different λ values. We find that the value of UNK is gradually smaller when λ keeps increasing, but the result of OS* gradually increases, in addition, HOS first increases and then decreases. This shows that the classification effect of the adversarial network on the known classes is reduced due to the presence of unknown samples, and when we increase the weight of adversarial learning in the whole network, i.e., when we increase the value of λ, our effect on the known classes is improved.

Conclusion

In this paper, we proposed a two-stage method called EOS. This method leverages an energy-based separation strategy to tackle the OSDA problem. In the first stage, we enhanced the traditional classifier by introducing energy functions, allowing for the identification of unknown samples with high confidence and the generation of pseudo-labels through secondary separation. In the second stage, we employed an adversarial network with instance-level weights and alternated the training of the classifier to refine the separation results. This dynamic weighted synergistic mechanism enables the separation and alignment networks to co-evolve through continuous alternating training, allowing the model to adapt to shifting feature distributions across domains. The integration of these two stages forms a robust, end-to-end solution for open-set domain adaptation. Our experiments on multiple datasets demonstrate that the model is highly effective across datasets with varying degrees of domain

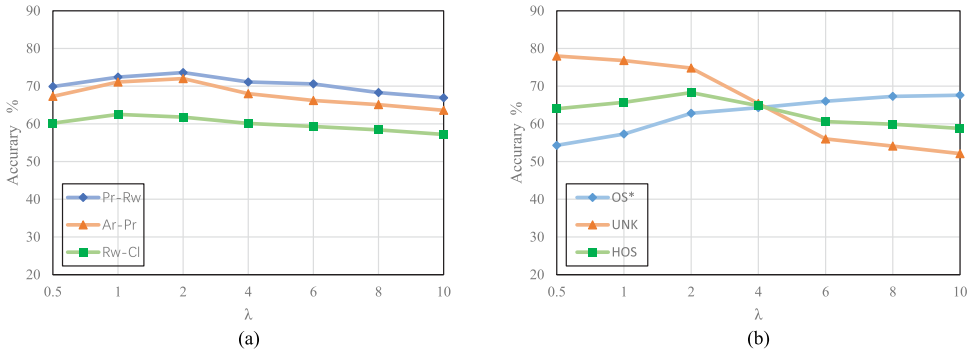


Fig. 4 Parameter sensitivity analysis: **a** shows the variation of HOS metrics following lambda for the three subtasks on the Office-Home dataset. **b** shows the three metrics on the Cl-Rw task when different lambda values are taken

divergence, showcasing strong robustness and adaptability. In the future, we aim to address the limitation that pseudo-label quality heavily relies on the accuracy of the initial separation. To improve this, we plan to explore leveraging pre-trained models like CLIP for generating more reliable pseudo-labels. Additionally, we will work on extending the EOS model for source-free open-set domain adaptation to enhance its applicability in real-world scenarios.

Acknowledgements We gratefully acknowledge the support and assistance provided by Sifan Long, Zhiwei Hao, and Yiyang Li during the early stages of this manuscript's development.

Author Contributions ZF: writing-review & editing, conceptualization, methodology, software, validation, visualization, data curation. DL: writing-review & editing, project administration, formal analysis, methodology, conceptualization, supervision, funding acquisition. SW: writing-original draft, conceptualization, methodology, formal analysis, validation, project administration, supervision. HC: writing-original draft, formal analysis, conceptualization, methodology.

Funding This research was funded by National Natural Science Foundation of China (No. 62376106), Natural Science Foundation of Hunan Province (No. 2023JJ50392), Scientific Research Fund of Hunan Provincial Education Department (No. 23A0588, 22C0564) and Aid Program for Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province.

Data Availability The dataset in this study is publicly available. The code can be shared upon reasonable request by contacting the corresponding author.

Declarations

Conflict of interest The author declares that there exists no Conflict of interest or personal relationship with other people and organizations regarding the publication of this article.

References

1. Wang Q, Meng F, Breckon TP (2024) Progressively select and reject pseudo-labelled samples for open-set domain adaptation. IEEE Transactions on Artificial Intelligence https://doi.org/10.48550/arXiv.2110.12635
2. Li P, Ni Z, Zhu X, Song J (2023) Distribution matching and structure preservation for domain adaptation. Complex & Intelligent Systems 9(2):1823–1835. https://doi.org/10.1007/s40747-022-00887-3
3. Wang J, Chen X, Zhang X-L (2024) Zeroth-and first-order difference discrimination for unsupervised domain adaptation. Complex & Intelligent Systems 10(2):2569–2584. https://doi.org/10.1007/s40747-023-01283-1
4. Busto PP, Gall J (2017) Open set domain adaptation, 754–763 https://doi.org/10.1109/ICCV.2017.88
5. Tzeng E, Hoffman J, Zhang N, Saenko K, Darrell T (2014) Deep domain confusion: Maximizing for domain invariance. CoRR https://doi.org/10.48550/arXiv.1412.3474 arXiv:1412.3474
6. Long M, Cao Y, Wang J, Jordan MI (2015) Learning transferable features with deep adaptation networks 37:97–105
7. Tang S, Chang A, Zhang F, Zhu X, Ye M, Zhang C (2024) Source-free domain adaptation via target prediction distribution searching. International journal of computer vision 132(3):654–672
8. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets, 2672–2680 https://doi.org/10.48550/arXiv.1406.2661
9. Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky VS (2016) Domain-adversarial training of neural networks. J. Mach. Learn. Res. 17:59–15935. https://doi.org/10.48550/arXiv.1505.07818
10. Xiao Z, Wang H, Jin Y, Feng L, Chen G, Huang F, Zhao J (2024) Spa: a graph spectral alignment perspective for domain adaptation. Advances in Neural Information Processing Systems 36;
11. Qu X, Liu L, Zhu L, Nie L, Zhang H (2024) Source-free style-diversity adversarial domain adaptation with privacy-preservation for person re-identification. Knowledge-Based Systems 283:111150
12. He J, Wu L, Tao C, Lv F (2024) Source-free domain adaptation with unrestricted source hypothesis. Pattern Recognition 149:110246

13. Saito K, Yamamoto S, Ushiku Y, Harada T (2018) Open set domain adaptation by backpropagation 11209:156–171. https://doi.org/10.1007/978-3-030-01228-1_10
14. Liu H, Cao Z, Long M, Wang J, Yang Q (2019) Separate to adapt: Open set domain adaptation via progressive separation, 2927–2936 https://doi.org/10.1109/CVPR.2019.00304
15. Bucci S, Loghmani MR, Tommasi T (2024) On the effectiveness of image rotation for open set domain adaptation 12361:422–438 https://doi.org/10.1007/978-3-030-58517-4_25
16. Zheng J, Wen Y, Chen M, Yuan S, Li W, Zhao Y, Wu W, Zhang L, Dong R, Fu H (2024) Open-set domain adaptation for scene classification using multi-adversarial learning. ISPRS Journal of Photogrammetry and Remote Sensing 208:245–260
17. Arbel M, Zhou L, Gretton A (2021) Generalized energy based models https://doi.org/10.48550/arXiv.2003.05033
18. Che T, Zhang R, Sohl-Dickstein J, Larochelle H, Paull L, Cao Y, Bengio Y (2020) Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling https://doi.org/10.48550/arXiv.2003.06060
19. Zhao JJ, Mathieu M, LeCun Y (2016) Energy-based generative adversarial network. CoRR arXiv:1609.03126 https://doi.org/10.48550/arXiv.1609.03126
20. Grathwohl W, Wang K, Jacobsen J, Duvenaud D, Norouzi M, Swersky K (2020) Your classifier is secretly an energy based model and you should treat it like one https://doi.org/10.48550/arXiv.1912.03263
21. Wang Q, Meng F, Breckon TP (2021) Progressively select and reject pseudo-labelled samples for open-set domain adaptation. CoRR arXiv:2110.12635 https://doi.org/10.48550/arXiv.2110.12635
22. Lecun Y, Chopra S, Hadsell R, Ranzato M, Huang F (2006) A tutorial on energy-based learning
23. Wang Y, Li B, Che T, Zhou K, Liu Z, Li D (2021) Energy-based open-world uncertainty modeling for confidence calibration. CoRR arXiv:2107.12628 https://doi.org/10.48550/arXiv.2107.12628
24. Kim T, Bengio Y (2016) Deep directed generative models with energy-based probability estimation. CoRR arXiv:1606.03439 https://doi.org/10.48550/arXiv.1606.03439
25. Raginsky M, Rakhlin A, Telgarsky M (2017) Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis 65:1674–1703 https://doi.org/10.48550/arXiv.1702.03849
26. Grandvalet Y, Bengio Y (2005) Semi-supervised learning by entropy minimization, 281–296
27. Cai G, Wang Y, He L, Zhou M (2020) Unsupervised domain adaptation with adversarial residual transform networks. IEEE Trans. Neural Networks Learn. Syst. 31(8):3073–3086. https://doi.org/10.1109/TNNLS.2019.2935384
28. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains 6314:213–226 https://doi.org/10.1007/978-3-642-15561-1_16
29. Venkateswara H, Eusebio J, Chakraborty S, Panchanathan S (2017) Deep hashing network for unsupervised domain adaptation, 5385–5394 https://doi.org/10.1109/CVPR.2017.572
30. Peng X, Usman B, Kaushik N, Hoffman J, Wang D, Saenko K (2017) Visda: The visual domain adaptation challenge. CoRR arXiv:1710.06924 https://doi.org/10.48550/arXiv.1710.06924
31. Jain LP, Scheirer WJ, Boult TE (2014) Multi-class open set recognition using probability of inclusion 8691:393–409 https://doi.org/10.1007/978-3-319-10578-9_26
32. You K, Long M, Cao Z, Wang J, Jordan MI (2019) Universal domain adaptation, 2720–2729 https://doi.org/10.1109/CVPR.2019.00283
33. Pan Y, Yao T, Li Y, Ngo C, Mei T (2020) Exploring category-agnostic clusters for open-set domain adaptation, 13864–13872 https://doi.org/10.1109/CVPR42600.2020.01388
34. Fang Z, Lu J, Liu F, Xuan J, Zhang G (2021) Open set domain adaptation: Theoretical bound and algorithm. IEEE Trans. Neural Networks Learn. Syst. 32(10):4309–4322. https://doi.org/10.1109/TNNLS.2020.3017213
35. Xu Y, Chen L, Duan L, Tsang IW, Luo J (2023) Open set domain adaptation with soft unknown-class rejection. IEEE Trans. Neural Networks Learn. Syst. 34(3):1601–1612. https://doi.org/10.1109/TNNLS.2021.3105614
36. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition, 770–778 https://doi.org/10.1109/CVPR.2016.90