Source-Free Active Domain Adaptation via Energy-Based Locality Preserving Transfer

Xinyao Li
University of Electronic Science and Technology of China
Chengdu, China
xinyao326@outlook.com

Zhekai Du
University of Electronic Science and Technology of China
Chengdu, China
zhekaid@std.uestc.edu.cn

Jingjing Li*
University of Electronic Science and Technology of China; Institute of Electronic and Information Engineering of UESTC in Guangdong
Chengdu; Dongguan, China
lijin117@yeah.net

Lei Zhu
Shandong Normal University
Jinan, China
leizhu0608@gmail.com

Ke Lu
University of Electronic Science and Technology of China
Chengdu, China
kel@uestc.edu.cn

ABSTRACT

Unsupervised domain adaptation (UDA) aims at transferring knowledge from one labeled source domain to a related but unlabeled target domain. Recently, active domain adaptation (ADA) has been proposed as a new paradigm which significantly boosts performance of UDA with minor additional labeling. However, existing ADA methods require source data to explicitly measure the domain gap between the source domain and the target domain, which is restricted in many real-world scenarios. In this work, we handle ADA with only a source-pretrained model and unlabeled target data, proposing a new setting named source-free active domain adaptation. Specifically, we propose a Locality Preserving Transfer (LPT) framework which preserves and utilizes locality structures on target data to achieve adaptation without source data. Meanwhile, a label propagation strategy is adopted to improve the discriminability for better adaptation. After LPT, unique samples with insignificant locality structure are identified by an energy-based approach for active annotation. An energy-based pseudo labeling strategy is further applied to generate labels for reliable samples. Finally, with supervision from the annotated samples and pseudo labels, a well adapted model is obtained. Extensive experiments on three widely used UDA benchmarks show that our method is comparable or superior to current state-of-the-art active domain adaptation methods even without access to source data.

CCS CONCEPTS

• **Computing methodologies** → **Active learning settings**.

*Corresponding author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548152

KEYWORDS

transfer learning, active learning, source-free domain adaptation, energy-based model

ACM Reference Format:
Xinyao Li, Zhekai Du, Jingjing Li, Lei Zhu, and Ke Lu. 2022. Source-Free Active Domain Adaptation via Energy-Based Locality Preserving Transfer. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548152

1 INTRODUCTION

The last decade has witnessed the astonishing success of deep learning. However, one outstanding problem is that current deep neural networks rely heavily on large amount of labeled training data to guarantee satisfactory performance [25, 41]. In real world multimedia applications, data distribution shift between different modalities is common [24, 34, 46], which prevents deep models from generalization. Considering that collecting labeled target data is time-consuming, expensive, and sometimes impossible, unsupervised domain adaptation [2, 41] (UDA), which aims to transfer knowledge from source domain to one relevant domain (target domain) with only unlabeled target data, has attracted much attention in the community.

Despite remarkable improvements, current UDA methods still fall far behind fully supervised methods in performance, largely limiting their practicability. Meanwhile, active learning [9, 32, 45] (AL) aims to actively label the most informative samples so that models with satisfactory performance can be obtained at an acceptable cost. In light of this, some recent efforts have been made to integrate AL into domain adaptation, which is known as active domain adaptation (ADA) [6, 35, 42]. The main challenge in ADA is that most target samples may be irrationally identified as informative by traditional AL methods, since domain discrepancy inherently exists between source and target domains, resulting in a large portion of target samples lying outside the source distribution. Therefore, the main challenge for ADA is two-fold: (i) *How to precisely select*

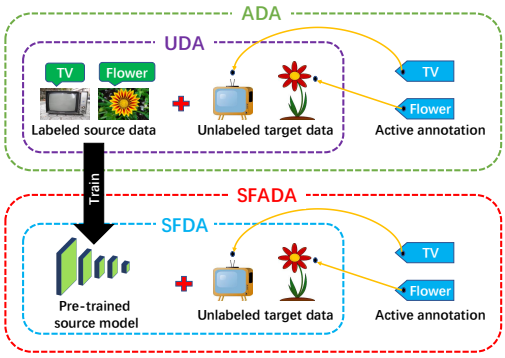


Figure 1: Differences and connections between unsupervised domain adaptation (UDA), active domain adaptation (ADA), source-free domain adaptation (SFDA) and source-free active domain adaptation (SFADA).

informative samples? and (ii) *How to fully exploit the information carried by these annotated samples?*

Existing ADA methods either learn a cross-domain query function for selecting active samples [6] or form clusters to acquire uncertain samples under domain shift [27]. However, they all assume the co-existence of source and target data during adaptation, which is impractical in many cases due to privacy or transmission issues, etc. Considering a medical auxiliary diagnostic system, where several hospitals jointly train the model but may not be willing to contribute their original patient data. For these source data-restricted scenarios, a frequently studied treatment is source-free domain adaptation (SFDA) [18, 20, 44] aiming to transfer knowledge from source to target with only a source-pretrained model and unlabeled target data. In this paper, we aim to tackle the source data-restricted problem in ADA, which we call source-free active domain adaptation (SFADA). The differences and connections among UDA, ADA, SFDA and SFADA are illustrated in Fig.1.

Traditional ADA methods fail to tackle the SFADA since the domain divergence is agnostic without the support of source data, which inevitably increases the difficulty to select the most informative target samples. On the other hand, it poses a new challenge in addition to the conventional ADA, i.e., (iii) *How to adapt the model to target domain when there exists no source samples?* In this work, we propose a framework that handles the challenge (iii) by leveraging the intrinsic properties hidden in target data structures. Specifically, we propose a strategy named Locality Preserving Transfer (LPT) that learns effective feature representations where the locality structure of target data is well-preserved thus forming clear clusters. Furthermore, identical label is propagated in each cluster to ensure that samples within each cluster have similar outputs, so as to improve the accuracy on target data. Inspired by manifold learning [43], we achieve the goal by constructing undirected graphs that characterize the certain statistical or geometrical properties of the dataset [43]. LPT is expected to gather unequivocal target samples into clear clusters, while for certain unique samples that share little locality with other in-class samples, they may not be well clustered and show poor discriminability. These are the samples we would like to actively annotate (i.e., challenge (i)).

However, how can we identify these poorly clustered samples? In this paper, we utilize Energy-Based Models (EBMs) [13] which have been successfully used for out-of-distribution (OOD) detection [8, 21]. Concretely, EBMs are capable of recognizing which samples are well handled by the current model and which are not, being a natural solution of challenge (i). Given corresponding output, the free energy of a sample can be derived from any trained classifier [8], which indicates the negative likelihood of being in-distribution. Usually, lower free energy means that current model is more confident about the output for the sample and vice versa. Intuitively, we tend to choose samples with higher energy for active annotation after clustering. Inspired by [20], we further introduce an energy-based pseudo labeling strategy. The pseudo labels of low energy samples are considered to have high accuracy, which along with the actively labeled samples further serve to supervise the remaining training process, ensuring the model can learn from unique samples while maintaining accuracy on unequivocal samples.

In general, the contribution of this paper can be summarized as: (1) We propose a new setting for UDA, i.e., source-free active domain adaptation (SFADA), which overcomes the performance deficiency of traditional UDA with minor additional cost while introducing no requirement for original source data, making it available in data-sensitive scenarios. (2) We propose an effective method for SFADA named Energy-based Locality Preserving Transfer (ELPT), which preserves and utilizes the inherent target data locality structure for adapting knowledge without source data. An energy-based strategy is further used for selecting unique and informative target samples. Besides, an energy-based pseudo labeling strategy is introduced for supervising training process. (3) We conduct extensive experiments on three mainstream UDA datasets. Experimental results show that our method is comparable or even superior to state-of-the-art ADA methods, which illustrates the efficacy of our method since we need no source data.

2 RELATED WORK

Unsupervised Domain Adaptation aims to train a model that works well on target domain with labeled source data and unlabeled target data [2, 17]. Current mainstream methods focus on mitigating domain gap between the two, which can be roughly divided in two types: Discrepancy-based [5, 14, 22, 23, 38] and adversarial-based [7, 15]. Discrepancy-based methods explore various kinds of metrics that best measure the distribution shift between the two domains and try to explicitly reduce it. Tzeng *et al.* [38] first propose to reduce domain shift by minimizing a metric named maximum mean discrepancy (MMD), which soon became one of the most frequently used metrics. Li *et al.* [14] propose to minimize inter-domain divergence and maximize intra-domain density by optimizing maximum density divergence (MDD). Adversarial-based methods normally introduce a domain discriminator for min-max game against the feature extractor, as described in [7]. Li *et al.* [15] further argue that the ability to defeat against generated confusing samples enhance the model's robustness and performance on target domain. However, all the traditional UDA methods require access to source data, which may not be possible in certain cases.

Source-free Domain Adaptation. In order to better handle situations where only a model trained on source data is available instead of source data, researches about source-free domain adaptation (SFDA) [19] have emerged. Liang *et al.* [20] perform mutual

information optimization on the pre-trained source model with classifier fixed while applying pseudo label strategy for target domain adaptation, which achieves impressive results. Yang *et al.* solve the SFDA problem by exploiting intrinsic neighborhood structures on the target domain. Li *et al.* [18] generate labeled target samples identical to real target samples by adversarial training, which is used for supervising training on the target domain.

Active Domain Adaptation. Active learning [1, 33] aims to find the most unique and informative samples for active annotation. Models trained on these samples are expected to show satisfactory performance with a low annotation cost. Considering annotated samples are scarce in UDA, active learning is desirable and some active UDA studies have emerged recently. [28, 31] are among the earliest studies towards active domain adaptation leveraging shallow methods. Fu *et al.* [6] propose three selection criteria that work together to decide the most informative target samples to be labeled. Considering potential clustering structures on target domain, Prabhu *et al.* [27] reveal informative target samples by embedding-based weighted clustering. Noticing that data distribution gap can be finely described by free energy [13], Xie *et al.* [42] present an energy-based method that effectively selects unique target samples and reduces domain discrepancy by explicitly minimizing energy gap between source and target.

3 METHOD

In source-free active domain adaptation, we only have access to an unlabeled target domain $\mathcal{D}_t = \{x_t\}$ and a model G pretrained on the source domain. We are also allowed to query b target labels, where b is quite small compared to total sample counts in $\mathcal{D}_t$. The goal of ELPT is to adapt the pretrained model G to the target domain for correctly classifying target samples.

3.1 Energy-Based Models

We first briefly revisit Energy-Based Models [13] which serve as a basic component in our method. EBMs essentially encode the dependence between two variables by assigning a scalar (energy) to each pair of them. For any discriminative model G that gives a discrete output y for every input x, we can define an energy function $E(x, y)$ with the following properties: $E(x, y)$ is lower if model G considers input x is more compatible to output y, that is, G is quite confident on assigning y to x. Consider a common scenario (also the scenario discussed in this paper) where model G is trained for classification tasks, every y consists of multiple logits $y_0, y_1, ..., y_{class\_num}$, representing how likely is it for G to classify x as the i-th class. A direct observation is that y_i works in a similar pattern with $E(x, y_i)$, the only difference is that y_i gets larger if x is more likely to belong to class i while $E(x, y_i)$ gives a smaller output. Based on this observation, we first define energy function for each logit as:

$$E(x, y_i) = -y_i, \tag{1}$$

where y_i is the i-th output for input x. With a collection of energies for data points, we can derive the joint distribution of x and y_i by Gibbs distribution:

$$p(x, y_i) = \frac{exp[-E(x, y_i)]}{Z}, \tag{2}$$

where $Z = \sum_x \sum_{i=1}^{c} exp[-E(x, y_i)]$ is a normalizing constant that calibrates energy values into probabilities between 0 and 1. Marginalizing over y_i, we can further obtain the marginal probability density for x:

$$p(x) = \sum_{i=1}^{c} p(x, y_i) = \frac{\sum_{i=1}^{c} exp[-E(x, y_i)]}{Z}. \tag{3}$$

However, one can hardly compute or reliably estimate Z [8] in practice. Therefore, we turn to computing free energy $E(x)$. Similar to $p(x)$, $E(x)$ can give us a parameterized estimation of the rationality of sample x. Similar to Eq.(2), we have:

$$p(x) = \frac{exp[-E(x)]}{Z}. \tag{4}$$

By connecting Eq.(1), Eq.(2) and Eq.(4), we have:

$$E(x) = -log \sum_{i=1}^{c} exp(y_i). \tag{5}$$

$E(x)$ provides a probability estimation with density proportional to $\exp(-E(x))$. It becomes larger for samples with lower probability (out-of-distribution) and vice versa, thus is suitable for selecting samples for active annotation.

3.2 Locality Preserving Transfer

Human beings have the instinctive ability to distinguish objects of different kinds at a glance [29]. The insight behind is that in human's visual spaces, objects in the same class typically have their locality structures. Although target samples may drop prediction accuracy with the source-trained model, the inherent data structure, however, is expected to be preserved [16]. Based on the observation, we first obtain the deep features via the feature backbone of the source-trained model. We then try to preserve this locality and make it more compatible with the target domain by improving the discriminability of features. Since our method is mainly based on locality preserving, we refer it as Locality Preserving Transfer (LPT).

As shown in Fig.2(a), the original state of target samples is chaotic and indistinguishable. The connected samples are of the same class but share little locality. Our goal is to find a feature space that well preserves the locality of samples, where in-class samples form clear and discriminative clusters, as shown in Fig.2(b). To ensure samples in the same cluster belong to the same class, a straightforward but efficient way is to ensure similar outputs among them. In other words, to propagate the label within each cluster by encouraging similar semantic outputs. Inspired by manifold learning [4], we reserve feature distances between samples by constructing an undirected graph W, where each data point is viewed as a vertex and the edge connecting vertexes i, j represents their distance as weight w_{ij}. Assume the model is composed of a feature extractor G with parameter θ_G as well as a classifier C with parameter θ_C. Classifier C takes into the extracted features $F = G(X)$ and gives semantic outputs $O = C(F)$. The feature extractor G is responsible for learning transferable feature representations. We measure feature distances by dot product, i.e., W is computed by:

$$W = FF^T. \tag{6}$$

We further build U that reserves output similarity between data points in the same way:

$$U = OO^T. \tag{7}$$

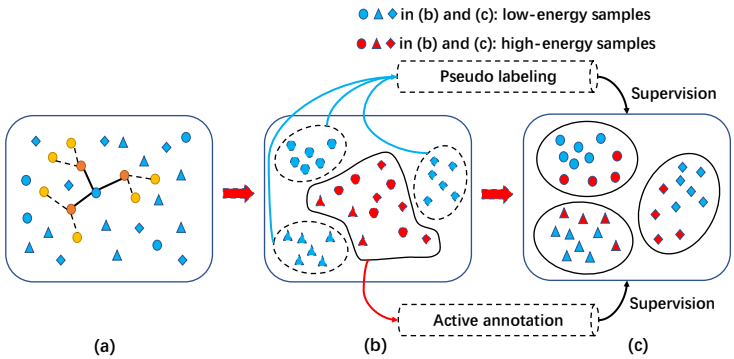


Figure 2: Idea illustration of our method. (a) illustrates how Locality Preserving Transfer (LPT) works. For each data point, we first find its K nearest neighbors ($K = 3$ in (a)), painted in orange. And for each of the K samples, we proceed to find M nearest neighbors of its ($M = 2$ in (a)), painted in yellow. We then encourage similar outputs from the K and M neighboring samples. LPT is performed on every sample to learn an effective feature space where data locality is well-preserved, resulting in (b), where most samples are well classified and form clusters. The misclassified red samples in (b) contain high energy while the blue ones are of low energy. We select red ones for active annotation and perform pseudo labeling on all samples. Pseudo labels of low energy samples (blue samples) together with actively annotated ones are used for supervising remaining training process, finally enabling the model to correctly classify all samples by forming more discriminative clusters, as shown in (c).

As discussed, we hope similar outputs in each cluster, that is to ensure the model to produce similar results for nearby samples. Therefore, for every data point (center point), we assign higher weight to its nearer neighbors, and maximize weighted sum of similarities between each center point and its neighbors. To avoid misguidance from far neighbors belonging to another cluster, we only consider K nearest neighbors (KNN) for each center point and promote similar outputs among them. We further collect M nearest neighbors for each of the K data points (M-KNN) and also provide supervision to center points but with a smaller constant weight w_m. The process is shown in Fig.2(a), where K is set to 3 and M is 2. To assure the weights are discriminative enough for neighbors in different distance, we first perform ℓ_2 normalization on features and compute weights of KNNs in an exponential way, i.e., $w^*_{ij} = exp(w_{ij}) - 1$. Accordingly, we modify the whole W by:

$$w^*_{ij} = \begin{cases} exp(w_{ij}) - 1, & \text{if } j \text{ is KNN of } i \\ w_m, & \text{if } j \text{ is M} - \text{KNN of } i. \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

With the modified similarity matrix W^* we define the similarity loss $\mathcal{L}_{sim}$ as the weighted sum of semantic similarities:

$$\mathcal{L}_{sim} = -UW^{*T}. \tag{9}$$

Note that our goal is to maximize the weighted sum of similarities among neighbors. By optimizing $\mathcal{L}_{sim}$, the data locality and prediction accuracy is simultaneously improved: closer data points are assigned higher weights, pushing them even closer and stabilizing the locality structure, while with higher weights, the model is more confident about assigning identical outputs among the near neighbors, contributing to higher accuracy.

3.3 Energy-Based Active Selection Strategy

As illustrated in Fig.2(b), LPT is expected to form several clusters by preserving data locality, allowing it to classify most unequivocal samples correctly. However, there are a few abnormal samples with unique or uncommon characteristics (red ones in Fig.2(b)), causing them to appear far from all other samples in the same class. In this case little information can be learned from their neighbors, because even the nearest neighbors are relatively far from them. These poorly clustered samples may further mislead other correct samples during LPT. Fortunately, Eq.(5) provides a natural way of selecting these hard-to-cluster samples. As stated in 3.1, free energy $E(x)$ is usually lower for in-distribution samples and higher for out-of-distribution ones, reflecting the performance gap between the two groups of samples. Thus, we propose the following strategy to actively select the unique samples:

Step 1: We first compute free energy for every data point and sort all samples according to free energy. We then select $\alpha\%$ samples with the highest free energies and form a candidate pool P_C.

Step 2: We notice that after LPT, the misclassified samples usually are those stay far from others, since every sample requires supervision from its KNNs and M-KNNs, and farther neighbors provide less weight thus less supervision. From this end, we select β samples with farthest KNNs from candidate pool P_C as final selection, forming an active sample pool P_A.

3.4 Energy-Based Pseudo Labeling

While actively selecting samples for annotation, we also apply pseudo labeling strategy to fully exploit the knowledge behind unlabeled data. Inspired by DeepCluster [3] and SHOT [20], we combine free energy with a weighted clustering to obtain centroids for each class. Considering that optimizing $\mathcal{L}_{sim}$ has pulled closer the samples with similar features and semantic outputs, it is natural to calculate the feature centroid for each class by weighting features according to their semantic outputs. Furthermore, since samples with higher free energy are with less reliable pseudo labels, we reduce the weight for these samples to avoid error diffusion. Specifically, we obtain the centroid c_k for class k by:

$$c_k = \frac{\sum_{i=1}^{n} \delta_k(C(G(x_i)))G(x_i)p_i}{\sum_{i=1}^{n} \delta_k(C(G(x_i)))}, \tag{10}$$

where δ_k is the k^{th} logit for semantic output after softmax, and parameter p_i reduces weight calculated from x_i:

$$p_i = \begin{cases} 0.1, & \text{if } \mathrm{E}(\mathrm{x_i}) > \text{thre} \\ 1, & \text{otherwise} \end{cases}. \tag{11}$$

$thre$ is dynamically adjusted over every epoch during training. With centroids for all classes, we can obtain the pseudo label $\hat{y_i}$ for every data point i by finding its closest centroid:

$$\hat{y_i} = \arg\min_k D(G(x_i), c_k), \tag{12}$$

where D is an arbitrary distance function, and *cosine* distance is used in this paper.

To avoid misguidance from wrong pseudo labels, we screen out samples with lower free energy for supervised training. Unlike high energy samples, as illustrated in Fig.2(b), low energy samples are better organized in clusters, forming neighboring structures in both feature and semantic level. Since our pseudo labeling method weights all features by their corresponding semantic output, these clustered samples tend to provide similar and strong supervision while calculating centroids. In other words, they affect the position of centroids to a more than the incompact high energy samples. As a result, low energy samples stay closer to centroids of corresponding classes, thus are more likely to have correct pseudo labels. Extensive experiments in 4.4 further support our assumption.

The pseudo labeling is conducted several times parallel to active annotation during training. We propose to use a larger proportion μ_0 of pseudo labels at the beginning, and less but more accurate ones as the training proceeds. The reason is that, LPT tends to learn the most general features and locality structures at the beginning. Applying more pseudo labels at the beginning promotes faster convergence rate and higher robustness even if the accuracy of pseudo labels is lower. As the training proceeds, the model learns more specific information from the actively annotated samples, then we gradually reduce the number of pseudo labels for higher accuracy, because these newly learned information is vulnerable to misguidance from wrong pseudo labels. Assume the samples with reliable pseudo labels form a pseudo label pool P_L, we have the following classification loss $\mathcal{L}_{clf}$:

$$\mathcal{L}_{clf} = \ell_{ce}(\mathcal{X}_P, \mathcal{Y}_P), \tag{13}$$

where $\mathcal{X}_P \in P_A \cup P_L$. $\ell_{ce}(\cdot,\cdot)$ stands for the standard cross entropy loss, and $\mathcal{Y}_P$ are corresponding actively annotated labels (for $\mathcal{X} \in P_A$) or pseudo labels (for $\mathcal{X} \in P_L$).

The active annotation and pseudo labeling process in fact cooperatively promotes each other. With supervision from active samples, the model learns to adjust feature space so that these previously hard-to-cluster samples can now find appropriate neighbors. These new members in clusters provide new information for centroid computation in turn, updating the centroids for more robust pseudo labeling performance.

3.5 Training Process

We further apply the frequently used diversity loss [20, 37, 44] to prevent trivial solutions. To be more specific, the solutions may not be diverse enough so that some minor classes are hardly involved.

$$\mathcal{L}_{div} = \sum_{i=1}^{c} \hat{p_i} \log \hat{p_i}, \tag{14}$$

Algorithm 1 Training Algorithm of ELPT

Require: A model $\mathcal{G}$ with parameters $\theta_{\mathcal{G}}$ pretrained on source data, unlabeled target data $\mathcal{X}_t$, active annotation budget β, μ and Δ_μ for deciding the proportion of pseudo labels used.

```
while target locality structure not preserved do
    Compute L_sim, L_div by Eq.(9), Eq.(14).
    Update θ_G.
end while
Set number of already annotated samples α = 0.
while α < β do
    Select α* informative samples x_a from X_t following selection strategy stated in 3.3.
    Obtain annotated labels y_a.
    Obtain pseudo labels of X_t following the pseudo labeling strategy in 3.4.
    Select μ% samples x_p with the lowest free energy and their corresponding pseudo labels y_p.
    Form training data X_train = x_a ∪ x_p, Y_train = y_a ∪ y_p.
    for i in max_iter do
        Calculate L_clf by Eq.(13).
        Update θ_G.
    end for
    Update α = α + α*, μ = μ − Δ_μ.
end while
return Parameters of trained model θ_G.
```

where $\hat{p_i}$ is mean value of output for class i over all samples. Minimizing $\mathcal{L}_{div}$ helps promote the diversity of output over all classes, thus avoiding trivial solutions. $\mathcal{L}_{div}$ is optimized along with $\mathcal{L}_{sim}$ during LPT.

As a result, the overall learning loss can be expressed as:

$$\mathcal{L}_1 = \mathcal{L}_{sim} + \mathcal{L}_{div}, \ \mathcal{L}_2 = \mathcal{L}_{clf}. \tag{15}$$

The losses are organized as $\mathcal{L}_1$ and $\mathcal{L}_2$, since they are not optimized simultaneously but in a two-step fashion. To summarize, we first perform LPT on the source-pretrained model with unlabeled target data for learning relatively general target features and locality structures, optimizing $\mathcal{L}_1$. And then we actively annotate hard samples for several rounds based on energy. For every round, pseudo labeling strategy is simultaneously performed. We then screen out samples with reliable pseudo labels for supervised training together with the actively annotated ones. Supervised training lasts for several epochs before next round of active annotation, optimizing $\mathcal{L}_2$. The training completes when annotation budget runs out. The overall training process is described in Algorithm 1.

4 EXPERIMENTS

4.1 Datasets

We perform extensive experiments on three widely-used domain adaptation benchmarks: Office31 [30], Office-home [40], and VisDA-2017 [26]. Office31 is a dataset with 31 classes of samples divided into 3 domains: Amazon, Dslr and Webcam. Office-home is a dataset with 65 classes and 4 domains: Art, Clipart, Product and RealWorld.

VisDA-2017 is a challenging large-scale dataset divided into 2 domains: Synthesis (containing 152k samples) and Real (containing 55k samples), covering 12 classes.

4.2 Implementation Details

We implement our method based on PyTorch 1.4.0 with Python version 3.7, and run all experiments on a RTX 2080Ti GPU.

Pretraining on source data. We use ResNet [10] pretrained on ImageNet [12] and replace the last classification layer by a bottleneck layer and a task-specific classification layer. We also adopt the Label Smoothing technique used in SHOT [20]. For all three datasets, SGD optimizer with momentum 0.9 is adopted for updating parameters and the batch size is set to 64. For Office31 and Office-home we train a Resnet-50 for 20 epochs with learning rate 1e-3 for feature extraction network and 1e-2 for the rest. For VisDA-2017, we train a Resnet-101 for 10 epochs with learning rate 1e-4 for feature extraction network and 1e-3 for the rest.

Training process of ELPT. For all datasets, we set learning rate to 1e-4 for feature extraction network and 1e-3 for the rest. The batch size is 64 for all datasets. We use SGD optimizer with momentum 0.9 for all tasks. Following previous active domain adaptation works [6, 42], the budget β for active annotation (β in Algorithm 1) is set to 5%. The numbers of KNN and M-KNN are both set to 2 for Office31 and Office-home, while for VisDA-2017 is 5 and 3 respectively. We repeat LPT for 20 epochs on Office31 and Office-home, and for VisDA-2017 only 2 epochs is conducted. Note that the w_{ij} in Eq.(8) is clamped to $[0.1, 1]$ to avoid unexpected weights caused by exponential operation. Then we proceed to select 1% samples each time (α^* in Algorithm 1) following active selection strategy in 3.3, and train for 5 epochs (max_iter in Algorithm 1) before next round of active annotation. For the first round, proportion of used pseudo labels is 75% and descends by 6.25% each round (μ and μ_Δ in Algorithm 1), reaching 50% in the end.

4.3 Results

To the best of our knowledge, there is no current research on source-free active domain adaptation, so we compare our method with various active domain adaptation [6, 27, 36, 42] and active learning [1, 11] methods. Note that our method requires no source data, which is a more challenging scenario than ADA. For further comparison, we provide experiment results of Source-only (directly apply the model pretrained on source data to target), LPT (accuracy after LPT, before active adaptation) and RAN (randomly select active samples). Experiment results on Office31, Office-home and VisDA-2017 are presented in Table 1 and Table 2 respectively.

On dataset **Office31**, ELPT obtains comparable mean accuracy with current state-of-the-art ADA methods. Concretely, on tasks A→D and A→W, we obtain significant improvement than the source only model, almost approaching fully-supervised level, which is quite considerable since we have no access to source data. On tasks D→A and W→A we are slightly behind EADA. Given the ordinary performance of LPT on the these 2 tasks, one possible reason is that domain Amazon (A) forms poor clustering structure and limits the performance of LPT, leaving too many misclassified hard samples for active annotation that largely exceed the 5% budget, thus eventually pulling down the overall accuracy.

Table 1: Accuracy on Office31 with 5% active labeling budget.

Methods	A→D	A→W	D→A	D→W	W→A	W→D	Mean
Source Only	80.9	74.8	62.6	96.2	63.4	98.4	79.4
AADA [36]	89.2	87.3	78.2	99.5	78.7	**100.0**	88.8
ADMA [11]	90.0	88.3	79.2	**100.0**	79.1	**100.0**	89.4
BADGE [1]	90.8	89.1	79.8	99.6	79.6	**100.0**	89.8
TQS [6]	92.8	92.2	80.6	**100.0**	80.4	**100.0**	91.1
CLUE [27]	92.0	87.3	79.0	99.2	79.6	99.8	89.5
EADA [42]	97.7	96.6	**82.1**	**100.0**	**82.8**	**100.0**	**93.2**
LPT	94.2	94.0	76.4	98.2	75.8	99.8	89.7
RAN	95.2	95.4	79.2	98.4	78.8	**100.0**	91.2
ELPT	**98.0**	**97.2**	81.2	99.4	80.7	**100.0**	92.8

On **Office-home**, we surpass all competing methods. Though in a small degree, the improvement still strongly proves the superiority of our method, since we require no source data but a pretrained source model. On tasks Cl→Ar and Cl→Rw, ELPT even considerably surpasses current state-of-the-art ADA methods by a large margin (more than 2%), indicating the effectiveness of LPT and active annotation strategy. We can observe that with only LPT we are able to surpass most active learning or even ADA methods. The solidness of LPT sets the foundation for further active adaptation. Another observation is that the lower accuracy LPT achieves, the more is learned from active adaptation. On the 3 tasks with lowest LPT accuracy (Ar→Cl, Pr→Cl, Rw→Cl), we can observe the most significant improvements (6.7%↑, 6.3%↑, 6.1%↑) upon LPT, which implies that learning from active samples also contributes to the pseudo labeling process and generates more accurate pseudo labels.

It is obvious that considerable improvement has been achieved on the most challenging dataset **VisDA-2017**. With the large amount of samples, LPT is able to utilize more information from more KNNs and M-KNNs, promoting higher accuracy and faster convergence rate. Active adaptation further enhances the model's ability to classify unique samples, finally resulting in a well adapted model.

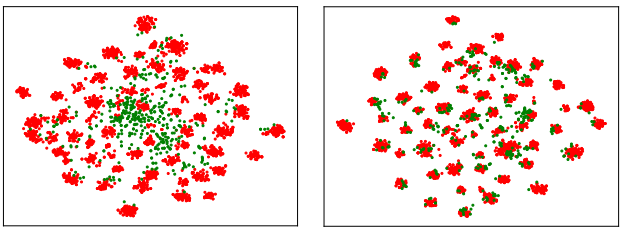
As shown in Table 1 and Table 2, benefiting from the favorable performance of LPT, we can achieve better results than most previous methods even by randomly selecting active samples. However, there is still a significant gap to ELPT, showing the necessity of our active selection strategy.

4.4 Analysis

Feature visualization. We use t-SNE [39] to visualize features from ResNet backbone to intuitively present data locality structures on different phases of our method. As shown in Fig.3, we render 70% lowest-energy samples in red and 10% highest-energy ones in green. We can observe from Fig.3(a) that the low-energy samples (red ones) have formed a general clustering structure, which, though not discriminative enough, still illustrates the efficacy of LPT. We also notice that these high-energy samples (green ones) well cover most misclassified and incompact samples, proving the correctness of identifying unique samples by free energy. Fig.3(a) well align with the expected result stated in 3.3 and Fig.2(b), which supports the soundness of LPT. After active annotation and remaining training process we are able to obtain a more discriminative locality structure shown in Fig.3(b). Another key observation is that most of these high-energy samples are correctly classified, indicating that the model indeed learns from actively annotated samples. Generally, the feature visualization illustrates that our training process is valid and effective.

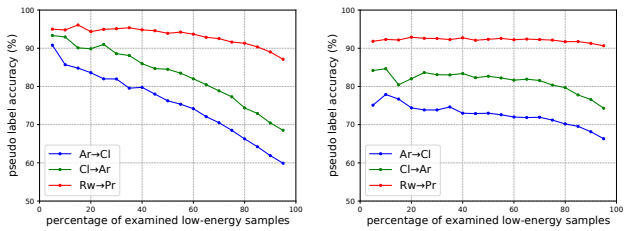
Table 2: Accuracy on Office-home and VisDA-2017 with 5% active labeling budget.

Methods	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean	**VisDA-2017**
Source Only	43.9	66.3	74.5	51.4	61.3	64.0	50.6	40.8	72.5	64.7	45.6	77.6	59.4	47.4±0.3
AADA [36]	56.6	78.1	79.0	58.5	73.7	71.0	60.1	53.1	77.0	70.6	57.0	84.5	68.3	80.8±0.4
ADMA [11]	57.2	79.0	79.4	58.2	74.0	71.1	60.2	52.2	77.6	71.0	57.5	85.4	68.6	81.4±0.4
BADGE [1]	58.2	79.7	79.9	61.5	74.6	72.9	61.5	56.0	78.3	71.4	60.9	84.2	69.9	84.3±0.3
TQS [6]	58.6	81.1	81.5	61.1	76.1	73.3	61.2	54.7	79.7	73.4	58.9	86.1	70.5	83.1±0.4
CLUE [27]	58.0	79.3	80.9	68.8	77.5	76.7	66.3	57.9	81.4	75.6	60.8	86.3	72.5	85.2±0.4
EADA [42]	63.6	**84.4**	83.5	70.7	83.7	80.5	**73.0**	**63.5**	85.2	**78.4**	65.4	88.6	76.7	88.3±0.1
LPT	58.6	79.5	80.3	67.2	78.9	77.2	64.1	57.0	81.6	70.5	59.5	84.9	71.6	84.1±0.3
RAN	63.2	82.0	81.6	70.3	81.8	80.0	67.0	62.1	83.5	72.8	63.6	86.6	74.5	87.5±0.3
ELPT	**65.3**	84.1	**84.9**	**72.9**	**84.4**	**82.8**	69.8	63.3	**86.1**	76.2	**65.6**	**89.1**	**77.0**	**89.2±0.3**



(a) After LPT (b) Final

Figure 3: The t-SNE visualization of task Art→Clipart on Office-home. Data points in red are low-energy ones and those in green are high-energy ones. Subfigure (a) and (b) show geometrical feature distribution after LPT and whole training procedure respectively.



(a) After LPT (b) Final

Figure 4: Relation between pseudo label accuracy and the percentage of chosen lowest-energy samples of 3 tasks on Office-home (Ar→Cl in blue, Cl→Ar in green, Rw→Pr in red). (a) shows the relation when LPT is completed, and (b) shows that when all training process is completed.

Table 3: Ablation study on VisDA-2017.

$\mathcal{L}_{sim}$	$\mathcal{L}_{clf}$	$\mathcal{L}_{div}$	Accuracy
			47.7
√			67.9
	√		83.3
√	√		75.6
√		√	84.1
√	√	√	**89.2**

Ablation study. We conduct ablation experiments to study how each module of our method contributes to the final result. The results can be found in Table 3. Specifically, the first conclusion is that the $\mathcal{L}_{div}$ loss is crucial to LPT. For the experiment solely on $\mathcal{L}_{sim}$ (the second row), we observe the degeneration problem stated in 3.5, where certain classes are hardly involved. Specifically, accuracies on classes bicycle, skateboard and truck are 0%. However, $\mathcal{L}_{sim}$ equipped with $\mathcal{L}_{div}$ (the fifth row) is able to reach 86.6%, 89.9% and 36.3% on these 3 classes. The reason for the phenomenon is that the model finds a trivial way to optimize $\mathcal{L}_{sim}$ by grouping the samples into fewer clusters (9) than actual class number (12). The problem also greatly pulls down the efficacy of $\mathcal{L}_{clf}$, resulting in lower accuracy on combination $\mathcal{L}_{sim}, \mathcal{L}_{clf}$ than pure $\mathcal{L}_{clf}$. Applying pure $\mathcal{L}_{clf}$ obtains 35.6% improvement than the source-only model (the first row), implying that the model learns far more than the 5% actively annotated samples, and the information carried by active samples is well exploited. Applying $\mathcal{L}_{clf}$ with $\mathcal{L}_{sim}$ helps alleviate the degeneration problem to some extent, gaining 7.7% improvement than pure $\mathcal{L}_{sim}$. In addition, free energy helps screen out samples from the three degenerated classes during pseudo labeling, and active annotation provides correct supervision for these classes, improving their accuracy to 6.9%, 29.3% and 27.6% respectively.

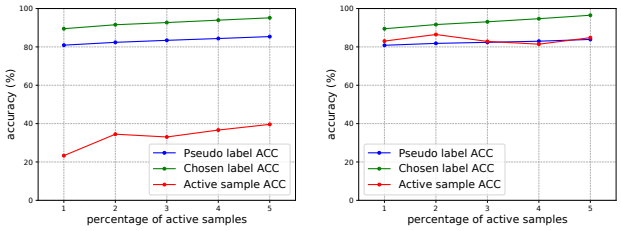
Accuracy of pseudo labels. Fig.4 shows how pseudo label accuracy changes with the proportion of chosen lowest-energy samples (parameter μ in Algorithm 1). When LPT is complete, a small number of uncertain, misclassified samples exist, and free energy emphasizes them well by assigning high values to them. As shown in Fig.4(a), examining only 5% of the lowest-energy samples gives us over 90% pseudo label accuracy, which drops evenly with increased examining proportion. The observation supports that by selecting lowest-energy samples we do obtain more accurate pseudo labels. When all training process is complete, as shown in Fig.4(b), the accuracy curves remain still for most choices of examine proportion, indicating that the model has learned most information provided by active samples, and there is little left to exploit by free energy. In other words, the model now treat most samples as general and well-classified ones. The inaccuracy in this stage comes from the small proportion of wrong pseudo labels used.

Fig.5 shows accuracy of all pseudo labels (blue), accuracy of selected labels used for training (green) and accuracy of pseudo labels of chosen active samples (red). We expect to annotate previously misclassified samples so ideally, the pseudo label accuracy of chosen active samples should be low. In Fig.5(a), the red curve indicates that samples selected by our strategy are mostly misclassified, while randomly selecting active samples (Fig.5(b)) results in annotating large proportion of originally correct samples, which hardly contributes to the training process. The pseudo label accuracy of samples selected for supervising training (green curve) is constantly higher than accuracy over all pseudo labels, which minimizes the potential misguidance from wrong pseudo labels.

Performance under different active budgets. We further perform experiments on different active budgets to demonstrate the scalability of our method. Given the fact we have already reached over 89% accuracy using the ResNet-101 backbone with only 5% budget, there may be little room to improve with more budget. Hence, we adopt the ResNet-50 backbone for the experiment. Three other methods and the random selection is compared with ELPT.

Table 4: Accuracy on Office-home under different K and M with 5% active labeling budget.

	Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean
K=1,M=1	LPT	52.3	72.7	79.0	60.8	73.2	76.1	60.0	50.5	80.1	67.6	54.9	80.1	67.3
	ELPT	60.5	80.9	84.8	69.1	81.6	**83.3**	68.2	58.6	85.3	74.9	62.7	88.0	74.8
K=2, M=1	LPT	57.8	79.2	81.0	65.8	77.0	77.4	65.3	56.6	82.0	71.1	58.5	83.5	71.3
	ELPT	64.1	84.0	**85.4**	71.9	83.3	83.1	**70.9**	63.1	**86.4**	**76.2**	65.1	88.8	76.9
K=2, M=2	LPT	58.6	79.5	80.3	67.2	78.9	77.2	64.1	57.0	81.6	70.5	59.5	84.9	71.6
	ELPT	**65.3**	**84.1**	84.9	**72.9**	**84.4**	82.8	69.8	**63.3**	86.1	**76.2**	**65.6**	89.1	**77.0**
K=3, M=2	LPT	58.0	80.0	80.4	67.2	77.2	75.4	63.7	56.4	78.8	70.3	59.2	84.4	70.9
	ELPT	64.2	84.0	84.7	**72.9**	82.0	80.7	69.4	62.7	83.3	75.2	65.0	88.1	75.5
K=3, M=3	LPT	56.3	79.0	80.1	64.8	75.6	75.0	63.5	56.1	78.5	71.1	57.9	85.2	70.3
	ELPT	62.0	83.9	84.7	70.5	81.0	80.6	69.2	62.3	83.3	76.1	64.3	**89.2**	75.6



(a) ELPT (b) Random

Figure 5: Accuracy of all pseudo labels, chosen labels and pseudo labels of active samples on VisDA-2017. Subfigure(a) shows our method while subfigure(b) shows result of randomly selecting active samples.

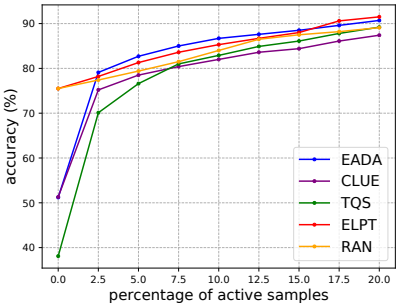


Figure 6: Accuracy under different active budget on VisDA-2017 using Resnet50 as backbone. Multiple methods are compared with ELPT.

As shown in Fig.6, the accuracy of ELPT increases steadily with increased budget, implying that in each active selection round ELPT is able to continuously select informative samples for active annotation. It is worth noting that for lower budget, our ELPT is slightly behind EADA. We conjecture the reason is that the shallower ResNet-50 provides weaker feature extraction ability compared with ResNet-101, limiting LPT to exploit more transferable data locality structure. Without source data, the data structure is even more prone to be destroyed. Therefore, ELPT should use more target semantic information to uncover the underlying data structure, thus consuming certain annotation budget to fill the source data gap to EADA. Eventually, ELPT is able to learn more specific target structure, reflecting in higher accuracy than EADA after consuming 15% budget.

Sensitivity to K and M. Table 4 shows how different K and M affect the performance on Office-home. We perform extensive experiments on different combination of K and M ranging from 1 to 3. The mean accuracy over all 12 tasks is mainly affected by K, since M-KNNs are always assigned smaller constant weight than KNNs. Among these choices, $K = 2$ gives the best performance, reaching a mean accuracy of 77.0%. $K = 1$ takes insufficient neighbors into consideration thus the locality structure on target is not fully preserved, bringing significantly lower mean LPT accuracy. Setting K=3 on the other hand, may occasionally receive misguidance from farther neighbors, which leads to less discriminative clustering structures.

For individual tasks, there are tasks insensitive to the variation of K and M (e.g., Ar→Rw, Rw→Ar, Rw→Pr) and also relatively sensitive ones (e.g., Ar→Cl, Cl→Pr). One may argue that it is time-consuming to find suitable K and M for each task, whereas our experiments show that by only fixing a constant K over all tasks, one can achieve similar overall performance to the best configuration. By considering the best individual task results shown in Table 4 (bold ones), we can get mean accuracy of 77.25%. The gap is marginal between results of simply setting $K = 2$ for all tasks, which indicates the robustness and practicability of our method.

5 CONCLUSION

In this paper, we integrate SFDA into ADA, forming a new setting source-free active domain adaptation. We further propose a novel framework termed ELPT for SFADA problems. Without source data, ELPT first explores and preserves target data locality with a graph-based approach called LPT. Based on this, an energy-based selection strategy is adopted for more informative active samples. Simultaneously, ELPT generates pseudo labels for all samples and effectively screens out those with high-accuracy pseudo labels. Finally, these samples together with active samples are used for supervising training process, promoting more specific adaptation. The two factors, i.e., target data structure and informative active samples, are major concerns of ELPT. They well fit source-free active domain adaptation problems, which is expected to inspire further progress in this new setting.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62176042 and 62073059, and in part by CCF-Baidu Open Fund (NO.2021PP15002000), and in part by CCF-Tencent Open Fund (NO. RAGR20210107), and in part by Guangdong Basic and Applied Basic Research Foundation (No. 2021B1515140013).

REFERENCES

[1] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671* (2019).

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning* 79, 1 (2010), 151–175.

[3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*. 132–149.

[4] Hwann-Tzong Chen, Huang-Wei Chang, and Tyng-Luh Liu. 2005. Local discriminant embedding and its variants. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 2. IEEE, 846–853.

[5] Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. 2021. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3937–3946.

[6] Bo Fu, Zhangjie Cao, Jianmin Wang, and Mingsheng Long. 2021. Transferable Query Selection for Active Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7272–7281.

[7] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[8] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. 2019. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263* (2019).

[9] Steve Hanneke et al. 2014. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning* 7, 2-3 (2014), 131–309.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Sheng-Jun Huang, Jia-Wei Zhao, and Zhao-Yang Liu. 2018. Cost-effective training of deep cnns with active model adaptation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1580–1588.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012).

[13] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

[14] Jingjing Li, Erpeng Chen, Zhengming Ding, Lei Zhu, Ke Lu, and Heng Tao Shen. 2020. Maximum density divergence for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3918–3930.

[15] Jingjing Li, Zhekai Du, Lei Zhu, Zhengming Ding, Ke Lu, and Heng Tao Shen. 2021. Divergence-agnostic Unsupervised Domain Adaptation by Adversarial Attacks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[16] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, and Heng Tao Shen. 2019. Locality preserving joint transfer for domain adaptation. *IEEE Transactions on Image Processing* 28, 12 (2019), 6103–6115.

[17] Jingjing Li, Mengmeng Jing, Hongzu Su, Ke Lu, Lei Zhu, and Heng Tao Shen. 2021. Faster domain adaptation networks. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. 2020. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9641–9650.

[19] Xinhao Li, Jingjing Li, Lei Zhu, Guoqing Wang, and Zi Huang. 2021. Imbalanced Source-free Domain Adaptation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3330–3339.

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

[21] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems* 33 (2020), 21464–21475.

[22] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

[23] Mingsheng Long, Jianmin Wang, Yue Cao, Jiaguang Sun, and S Yu Philip. 2016. Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering* 28, 8 (2016), 2027–2040.

[24] Xu Lu, Lei Zhu, Zhiyong Cheng, Liqiang Nie, and Huaxiang Zhang. 2019. Online Multi-modal Hashing with Dynamic Query-adaption. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 715–724.

[25] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[26] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. 2017. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924* (2017).

[27] Viraj Prabhu, Arjun Chandrasekaran, Kate Saenko, and Judy Hoffman. 2021. Active domain adaptation via clustering uncertainty-weighted embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8505–8514.

[28] Piyush Rai, Avishek Saha, Hal Daumé III, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*. 27–32.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

[30] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *European conference on computer vision*. Springer, 213–226.

[31] Avishek Saha, Piyush Rai, Hal Daumé, Suresh Venkatasubramanian, and Scott L DuVall. 2011. Active supervised domain adaptation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 97–112.

[32] Ozan Sener and Silvio Savarese. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489* (2017).

[33] Burr Settles. 2009. Active learning literature survey. (2009).

[34] Hongzu Su, Jingjing Li, Zhi Chen, Lei Zhu, and Ke Lu. 2022. Distinguishing Unseen From Seen for Generalized Zero-Shot Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7885–7894.

[35] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. 2020. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 739–748.

[36] Jong-Chyi Su, Yi-Hsuan Tsai, Kihyuk Sohn, Buyu Liu, Subhransu Maji, and Manmohan Chandraker. 2020. Active adversarial domain adaptation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 739–748.

[37] Hui Tang, Ke Chen, and Kui Jia. 2020. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8725–8735.

[38] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[39] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[40] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5018–5027.

[41] Mei Wang and Weihong Deng. 2018. Deep visual domain adaptation: A survey. *Neurocomputing* 312 (2018), 135–153.

[42] Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, Xinjing Cheng, and Guoren Wang. 2021. Active Learning for Domain Adaptation: An Energy-based Approach. *arXiv preprint arXiv:2112.01406* (2021).

[43] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and Stephen Lin. 2006. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE transactions on pattern analysis and machine intelligence* 29, 1 (2006), 40–51.

[44] Shiqi Yang, Joost van de Weijer, Luis Herranz, Shangling Jui, et al. 2021. Exploiting the Intrinsic Neighborhood Structure for Source-free Domain Adaptation. *Advances in Neural Information Processing Systems* 34 (2021).

[45] Donggeun Yoo and In So Kweon. 2019. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 93–102.

[46] Lei Zhu, Xu Lu, Zhiyong Cheng, Jingjing Li, and Huaxiang Zhang. 2020. Deep Collaborative Multi-View Hashing for Large-Scale Image Search. *IEEE Transactions on Image Processing* 29 (2020), 4643–4655.