# Selective Pseudo-Labeling with Reinforcement Learning for Semi-Supervised Domain Adaptation

Bingyu Liu[1]
liubingyu@bupt.edu.cn
Yuhong Guo[2,3]
yuhong.guo@carleton.ca
Jieping Ye[4,5]
jieping@gmail.com
Weihong Deng[1]
whdeng@bupt.edu.cn

[1] Beijing University of Posts and Telecommunications
Beijing, China
[2] Carleton University
Ottawa, Canada
[3] Canada CIFAR AI Chair, Amii
Edmonton, Canada
[4] University of Michigan
Ann Arbor, USA
[5] Beike
Beijing, China

**Abstract**

Recent domain adaptation methods have demonstrated impressive improvement on unsupervised domain adaptation problems. However, in the semi-supervised domain adaptation (SSDA) setting where the target domain has a few labeled instances available, these methods can fail to improve performance. Inspired by the effectiveness of pseudo-labels in domain adaptation, we propose a reinforcement learning based selective pseudo-labeling method for SSDA. It is difficult for conventional pseudo-labeling methods to balance the correctness and representativeness of pseudo-labeled data. To address this limitation, we develop a deep Q-learning model to select both accurate and representative pseudo-labeled instances. Moreover, motivated by large margin loss's capacity on learning discriminative features with little data, we further propose a novel target margin loss for our base model training to improve its discriminability. Our proposed method is evaluated on several benchmark datasets for SSDA, and demonstrates superior performance to all the comparison methods.

## 1 Introduction

Deep convolutional neural networks (CNNs) [7, 19] have achieved remarkable success in image classification tasks. When trained on large-scale labeled data, deep networks can learn discriminative representations and present great performance. However, it is difficult to collect and annotate datasets for many domains. A good option is to use the labeled data available in other domains for training models, which however often presents a domain shift challenge between the two domains and degrades the test performance. To address this

problem, many unsupervised domain adaptation (UDA) methods [4, 11, 17, 23] have been proposed. UDA aims to improve the generalization performance on unlabeled target domains. However, in reality a few labeled target instances can be available in target domains, and this semi-supervised domain adaptation (SSDA) setting is more common. According to [18], UDA methods often fail to improve performance compared with just training on the unified data of labeled source and target samples in the SSDA setting.

The purpose for domain adaptation is to improve the generalization performance in the target domain. In the SSDA setting, we have a few labeled target samples, but the number of them is too small to represent the distribution of target unlabeled data. To increase the number of labels in the target domain without incurring annotation cost, one intuitive strategy is to exploit pseudo-labels produced by a current prediction model. However, the pseudo-labels can often be very noisy and contain many wrong labels, while training with the mislabeled samples can negatively impact the original model. This motivates the straightforward selective pseudo-labeling strategy which selects the most confident pseudo-labels to increase their probability of correctness [1]. This simple strategy can select more accurate samples but non-necessary the most useful samples for the prediction model. For example, the more confident samples may be closer to the labeled data and fail to represent the distribution of the unlabeled data. It is more reasonable but difficult to perform selective pseudo-labeling by balancing the accuracy and the representativeness of the selected samples. To address this challenge, in this paper we propose a reinforcement learning based selective pseudo-labeling method. Our strategy is to use deep Q-learning to learn appropriate selection policies with reward functions that reflect both factors of label correctness and data representativeness.

In addition, due to the lack of labeled data in the target domain, the training methods typically have limited capacity in learning discriminative decision boundaries for the target domain. Inspired by the observation that large margin loss functions [2, 10, 21] can help learn discriminative features, we propose a contrastive target margin loss function over the labeled data from the source and target domains. As illustrated in Fig. 1, the decision boundary is mainly depending on source domain with the traditional softmax loss and the target margin loss can make the labeled target data play a bigger role in learning the decision boundary.

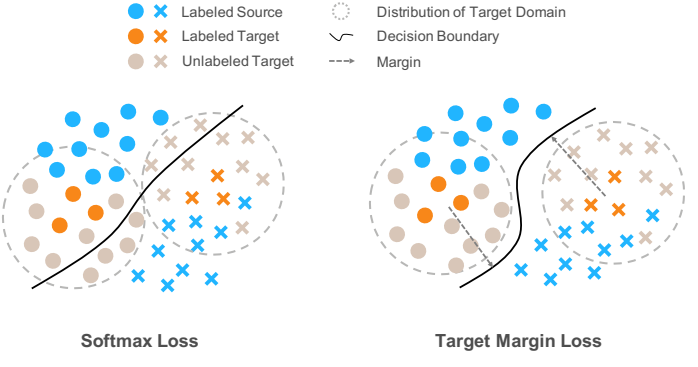


Figure 1: An illustration of the proposed target margin loss.

Overall, the contribution of this paper can be summarized as follows: (1) We propose a novel reinforcement learning framework for selective pseudo-labeling for semi-supervised domain adaptation. (2) We propose a contrastive target margin loss for SSDA that takes both generalization and discrimination into consideration. (3) Extensive experiments are

conducted on DomainNet [14], Office-31 [16] and Office-Home [20]. The results show that our proposed method achieves the state-of-the-art semi-supervised domain adaptation performance.

## 2 Related Work

**Domain Adaptation.** Domain adaptation [4, 23] aims to generalize models across different domains of different distributions. Most recent methods are focusing on unsupervised domain adaptation (UDA) which has a label-rich source domain and an unlabeled target domain. [4] proposes a domain-adversarial neural network (DANN) by adding a domain discriminator to minimize the distance between the feature distributions in source and target domain. [11] proposes Conditional Domain Adversarial Network (CDAN) by combining the discriminative information conveyed in the classifier predictions into the adversarial adaptation. In fact, semi-supervised domain adaptation (SSDA) is a more common setting in real-world datasets where a few labeled instances can be available. However, it has not been studied extensively, especially in the field of deep learning. A little conventional work [3, 25] has concentrated on this important task. As for deep learning based methods, [18] proposes a Minimax Entropy (MME) method by alternately maximizing the conditional entropy of unlabeled target data and minimizing it to optimize the classifier and the feature extractor respectively. Meanwhile, it shows that UDA methods can rarely improve accuracy in SSDA.

**Pseudo-Labeling.** Pseudo-labeling is an effective way to extend label set when the number of labels is limited. As for SSDA, pseudo-labeling can be used for target domain which has little labeled data. There are two strategies for pseudo-labeling without selection, hard labeling [22, 28] and soft labeling [13]. The hard labeling strategy assigns a pseudo-label with only one class predicted by the classifier to each unlabeled instance, which will be combined with original labeled data to train an improved model. However, due to the weak classifier in the initial stage of training, many samples will be mis-labeled. Using these mis-labeled data for supervised training can cause serious harm to the model. Thus, soft labeling has been employed by assigning the predicted conditional probability of all classes to the unlabeled data. As for selective pseudo-labeling [1, 24], a subset of unlabeled samples which are most confident in the prediction are selected to be pseudo-labeled. This sample selection strategy can make the pseudo-labels more accurate while it also has a limitation that the selected samples cannot represent the distribution of unlabeled data well.

**Reinforcement Learning.** Reinforcement learning (RL) is a technique which trains an agent to learn policies based on trial and error in a dynamic environment. The training strategy is to maximize the accumulated reward from the environment. RL has made great progress in many vision tasks. [26] designs an agent to label noisy web data so that right examples can be labeled to train a classifier. [27] proposes to choose sufficient data pairs for multi-shot person re-identification by training an agent. [9] introduces a policy network for adjusting a margin parameter in the loss function to learn more discriminative features from imbalanced face datasets. In this work, we train an agent with deep Q-learning to select more representative and accurate pseudo-labeled data.

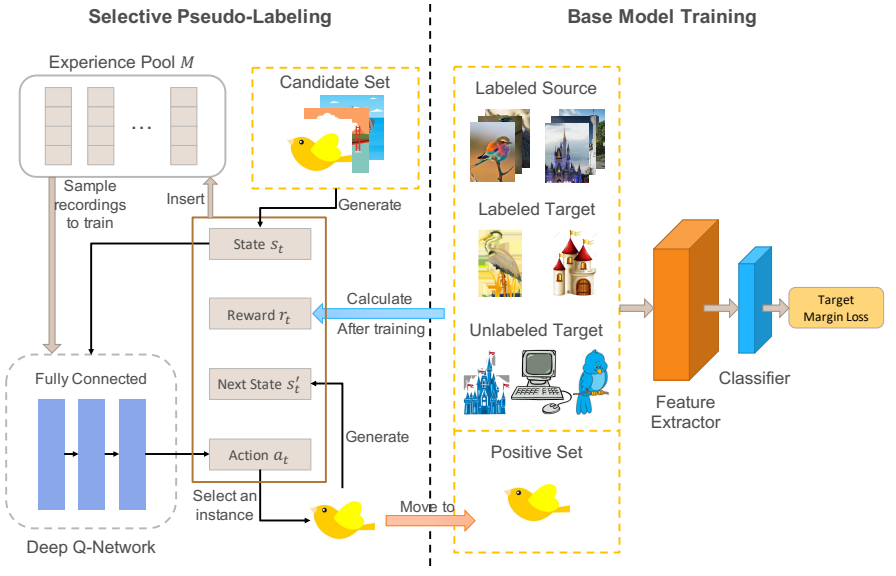


Figure 2: An overview of our method. We use target margin loss for the base model training and train an agent (deep Q-network) to select appropriate pseudo-labeled samples.

# 3 Method

For semi-supervised domain adaptation, we have a sufficient labeled dataset from the source domain, $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{N_s}$. In the target domain, we only have a limited number of labeled instances $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{N_t}$, but a large set of unlabeled instances $\mathcal{D}_u = \{(\mathbf{x}_i^u)\}_{i=1}^{N_u}$. The goal is to train a good prediction model on all these available data $\mathcal{D}_s$, $\mathcal{D}_t$ and $\mathcal{D}_u$ and evaluate it on $\mathcal{D}_u$ with the hidden true labels, as described in [18]. In this section, we present a novel selective pseudo-labeling method for SSDA. The method is based on a deep Q-learning framework and a target margin loss. The framework of the proposed method is depicted in Fig. 2. First, we use the proposed target margin loss to train a CNN consisting of a feature extractor $F$ and a classifier $C$ for a $K$-class classification problem. Then, we generate pseudo-labels for the unlabeled samples in the target domain based on the trained CNN classifier. Finally, we alternately train an agent with deep Q-learning and use the agent to select pseudo-labeled samples for the CNN training.

## 3.1 Target Margin Loss

Large margin loss functions [2, 10, 21] (based on traditional softmax loss function) effectively make CNN features more discriminative. For SSDA, there is a gap in feature distributions between domains and the number of labeled samples in target domain is much smaller than source domain. Therefore, we propose to add a margin to the loss on the target labeled data, by contrast to the loss on the source labeled data. This can be considered as making the decision boundaries more separated for the little target labeled data and representing the target distribution better. Then the proposed target margin loss can be formulated as follows:

$$\begin{aligned}\mathcal{L}_{tml} = &-\frac{1}{N_s}\sum_{i=1}^{N_s}\log\frac{e^{r\cos\left\langle F(\mathbf{x}_i^s),\mathbf{W}_{y_i^s}\right\rangle}}{\sum_{j=1}^{K}e^{r\cos\left\langle F(\mathbf{x}_i^s),\mathbf{W}_j\right\rangle}} \\ &-\frac{1}{N_t}\sum_{i=1}^{N_t}\log\frac{e^{r\cos\left(\left\langle F(\mathbf{x}_i^t),\mathbf{W}_{y_i^t}\right\rangle+m\right)}}{e^{r\cos\left(\left\langle F(\mathbf{x}_i^t),\mathbf{W}_{y_i^t}\right\rangle+m\right)}+\sum_{j=1,j\neq y_i^t}^{K}e^{r\cos\left\langle F(\mathbf{x}_i^t),\mathbf{W}_j\right\rangle}}\end{aligned} \tag{1}$$

where $\mathbf{W}_j$ denotes the $j$-th class weight of the last fully connected layer. $m$ is the margin parameter and $r$ is a re-scaling constant.

In addition, the suitable class separation can benefit from taking the unlabeled target data into account. To this end, we adopt the entropy loss to cluster the target unlabeled features into the corresponding decision regions, which is written as:

$$\mathcal{L}_{ent} = -\frac{1}{N_u}\sum_{i=1}^{N_u}\sum_{j=1}^{K} p\left(y=j|\mathbf{x}_i^u\right)\log p\left(y=j|\mathbf{x}_i^u\right) \tag{2}$$

where $p\left(y=j|\mathbf{x}_i^u\right)$ represents the probability of categorizing $\mathbf{x}_i^u$ to class $j$. Note that the samples should be clustered around the representative centers of the corresponding classes to decrease the entropy, resulting in the desired discriminative features.

Combining Eq. (1) and Eq. (2) provides the following formulation of our final semi-supervised loss function:

$$\mathcal{L} = \mathcal{L}_{tml} + \alpha\mathcal{L}_{ent} \tag{3}$$

where $\alpha$ is a hyper-parameter that balances the target margin loss and the entropy loss.

## 3.2 Selective Pseudo-Labeling by RL

Pseudo-labeling can generate mis-labeled samples which can cause serious harm to the subsequent learning process. To address this, selective pseudo-labeling is used in many works, which selects the most confident pseudo-labeled samples. This strategy however leads to the problem that the selected samples are generally easy ones or belong to easy classes. As a result, they cannot represent the target distribution well. Therefore, we propose to use reinforcement learning to select both representative and accurate pseudo-labeled samples.

We formulate the problem of selecting pseudo-labeled samples as a Markov Decision Process (MDP), described by states, actions and rewards, and train an agent to select pseudo-labeled samples. We define a candidate set $\mathcal{D}_c$ which contains pseudo-labeled samples to be selected and is initially randomly sampled from $\mathcal{D}_u$ with pseudo-labels, and a positive set $\mathcal{D}_p$ which contains the selected pseudo-labeled samples and initialized to be empty. During our CNN training, a series of reinforcement learning samples will be generated, which can be represented as $\{(s_i, a_i, r_i, s_i')\}$. Here, $s_i$ is the state and $r_i$ is the reward. $a_i$ is the action taken by the agent at state $s_i$, which equals to selecting a pseudo-labeled sample from the candidate set $\mathcal{D}_c$ and moving it to the positive set $\mathcal{D}_p$. $s_i'$ represents the next state which the agent turns to through the action $a_i$. Then, we alternately train the agent by using the reinforcement learning samples and use the agent to select pseudo-labeled samples for our CNN training.

**States.** We consider that the representative ability and accuracy of a pseudo-labeled target instance can be related to three parts, which are itself, the data with labels in $\mathcal{D}_t$ and $\mathcal{D}_p$, and the unlabeled data in $\mathcal{D}_u$. Note that pseudo-labeled instances are selected to make

the current distribution of labeled data close to the distribution of unlabeled data. Therefore, we formulate the state as a concatenation of three vectors, dependent on $\mathcal{D}_c$, $\mathcal{D}_t \cup \mathcal{D}_p$ and $\mathcal{D}_u$, respectively. For the first part, given the candidate set $\mathcal{D}_c = \{(\mathbf{x}_i^c, \hat{y}_i^c)\}_{i=1}^{N_c}$, we use a vector $\left[F(\mathbf{x}_i^c)^T, C(\mathbf{x}_i^c)^T\right] \in \mathbb{R}^{d+K}$ to represent each instance, where $F(\mathbf{x}_i^c)$ denotes the $d$-dimensional feature vector of instance $\mathbf{x}_i^c$ extracted by the feature extractor $F$, $C(\mathbf{x}_i^c)$ denotes the softmax output of the classifier $C$. Then we concatenate all the instances in $\mathcal{D}_c$ together. In addition, after moving an instance from $\mathcal{D}_c$ to $\mathcal{D}_p$, we replace the selected instance with a zero-valued vector. For the second part, we use a vector $\left[F(\mathbf{x}_i^{tp})^T, C(\mathbf{x}_i^{tp})^T\right] \in \mathbb{R}^{d+K}$ to represent each instance in $\mathcal{D}_t \cup \mathcal{D}_p$. Due to the available labels, we calculate the average in each class and then concatenate them together. The last part is represented by the instances in $\mathcal{D}_u$ with the same operation as the second part. Finally, the state $s_i$ is a flattened concatenation of these three parts.

**Actions.** For each state $s_i$, the agent takes an action $a_i$ to select the $a_i$-th instance in $\mathcal{D}_c$ and move it to $\mathcal{D}_p$. The number of actions is equivalent to the number of instances in $\mathcal{D}_c$.

**Rewards.** The rewards should reflect whether the actions taken by the agent are appropriate or not. In other words, selecting more representative and accurate pseudo-labeled instances should lead to positive rewards, and vice versa. We first define a metric function to measure the representative ability and accuracy, which can be formulated as follows:

$$\varphi(\mathbf{x}_i, \hat{y}_i) = \log p_c(y = \hat{y}_i|\mathbf{x}_i) + \beta \log p_f(y = \hat{y}_i|\mathbf{x}_i) + \lambda \Delta_e \tag{4}$$

where $\beta$ and $\lambda$ are hyper-parameters. $p_c(y = \hat{y}_i|\mathbf{x}_i)$ is the probability of the pseudo class $\hat{y}_i$ predicted by the classifier, which indicates the confidence of the prediction. We define $p_f(y = \hat{y}_i|\mathbf{x}_i)$ as:

$$p_f(y = \hat{y}_i|\mathbf{x}_i) = \frac{e^{s\cos\left\langle F(\mathbf{x}_i), \mathbf{z}_{\hat{y}_i}^{tp}\right\rangle}}{\sum_{j=1}^{K} e^{s\cos\left\langle F(\mathbf{x}_i), \mathbf{z}_j^{tp}\right\rangle}} \tag{5}$$

where $\mathbf{z}_j^{tp}$ denotes the feature center of the $j$-th class in target labeled and current positive set $\mathcal{D}_t \cup \mathcal{D}_p$, which can be calculated by:

$$\mathbf{z}_j^{tp} = \frac{\sum_{i=1}^{N_t} F(\mathbf{x}_i)\mathbb{I}(y_i = j) + \sum_{i=1}^{N_p} F(\mathbf{x}_i)\mathbb{I}(\hat{y}_i^p = j)}{\sum_{i=1}^{N_t} \mathbb{I}(y_i = j) + \sum_{i=1}^{N_p} \mathbb{I}(\hat{y}_i^p = j)} \tag{6}$$

$\mathbb{I}$ is the indicator function. Therefore, $p_f(y = \hat{y}_i|\mathbf{x}_i)$ is the softmax output of the cosine distance between $\mathbf{x}_i$ and the feature center of its pseudo class $\hat{y}_i$ in $\mathcal{D}_t \cup \mathcal{D}_p$.

The first two terms in Eq. (4) reflect the confidence of the pseudo-label prediction through two aspects, i.e., the output of the classifier and the similarity with the target feature center of the pseudo class. Since the classifier is more dependent on the source domain, we add the second term to take the distribution of the target domain into consideration. We also add a third term $\Delta_e = H - H'$, which denotes the decrease of the entropy of the target unlabeled data. $H$ and $H'$ represent the entropy at state $s_i$ and the next state $s_i'$ respectively and can be calculated in the same way as Eq. (2). In other words, we first calculate $H$ at state $s_i$ and then add a pseudo-labeled sample according to the action $a_i$ for one-epoch training. After the training, we calculate $H'$ and derive $\Delta_e$. In order to make the entropy not affected by $\mathcal{L}_{ent}$, we only use $\mathcal{L}_{tml}$ in Eq. (1) to optimize the model during the one-epoch training. Note that the more representative the selected sample is, the more the entropy will decrease.

**Algorithm 1** Selective pseudo-labeling by deep Q-learning

**Input:** $\mathcal{D}_s$, $\mathcal{D}_t$ and $\mathcal{D}_u$ **Output:** $F$, $C$ and $Q$

```
Pre-train F and C with D_s, D_t and D_u by using Eq. (3);
Initialize the positive set D_p = ∅, the experience pool M = ∅;
while not converge do
    Assign pseudo-labels to D_u by F and C and copy the parameters of F, C to F', C';
    Initialize D_c with random pseudo-labeled samples from D_u and generate the state s_0;
    while D_c ≠ ∅ and r_t > 0 do
        Get an output action a_t using Eq. (9) and update D_c and D_p by taking the action;
        Update F' and C' with D_s, D_t and D_p by optimizing the loss in Eq. (1);
        Generate the next state s'_t and calculate the reward r_t by Eq. (7) with F' and C';
        Insert the recording (s_t, a_t, r_t, s'_t) into M;
        Sample a batch of recordings {(s_i, a_i, r_i, s'_i)} from M to update Q by Eq. (8);
    end while
    Update F and C with D_s, D_t, D_u and D_p by optimizing the loss in Eq. (3).
end while
```

Therefore, larger $\Delta_e$ means stronger representative ability, and vice versa. In addition, we perform a log operation on the first two terms to keep these three terms at the same scale.

Directly using the metric function as reward can result in very small differences between the good and bad actions. Hence, we define the final reward function as follows:

$$r_i = \begin{cases} +1, & \varphi(\mathbf{x}_i, \hat{y}_i) > \tau \\ -1, & \varphi(\mathbf{x}_i, \hat{y}_i) \leq \tau \end{cases} \tag{7}$$

where $\tau$ is a threshold. We use this binary reward to provide the agent more explicit guidance.

**Deep Q-learning.** We apply deep Q-learning [12] to learn policies for selecting pseudo-labeled instances. For each state and action $(s_i, a_i)$, the output of the deep Q-network $Q(s_i, a_i)$ can represent the discounted accumulated rewards. Given a reinforcement learning training sample $(s_i, a_i, r_i, s'_i)$, the target value of $Q(s_i, a_i)$ can be calculated by $V_i = r_i + \gamma \max_{a'_i} Q(s'_i, a'_i)$, where $\gamma$ is a discount factor to decide the importance of future accumulated reward compared with the current reward. During the training, we iteratively update the Q-network by:

$$\Omega \leftarrow \Omega - \eta \sum_i \frac{\mathrm{d}Q(s_i, a_i)}{\mathrm{d}\Omega} (Q(s_i, a_i) - V_i) \tag{8}$$

where $\Omega$ represents the parameters of the Q-network. As for the entire model training, we alternately train the classification network and the Q-network. The details of our training strategy are summarized in Algorithm 1. When the agent (i.e., the Q-network) is called to select a pseudo-labeled instance at state $s_t$, it will output an action $a_t$ by a policy as follows:

$$a_t = \arg\max_a Q(s_t, a) \tag{9}$$

# 4 Experiments

## 4.1 Datasets and Baselines

**Datasets.** We perform our experiments on three datasets, DomainNet [14], Office-31 [16] and Office-Home [20]. Due to some noisy domains and categories in DomainNet, we pick

4 domains and 126 categories for DomainNet experiments. Following [18], we form 7 adaptation scenarios for testing with the 4 domains, Real (R), Clipart (C), Painting (P) and Sketch (S). As for Office-31, we construct 2 scenarios with Amazon (A) as the target domain since Webcam (W) and DSLR (D) do not have enough samples for effective evaluation. Office-Home is more difficult than Office-31 and contains four domains: Artistic (A), Clipart (C), Product (P) and Real-World (R). We randomly select three labeled samples per class as the labeled training target samples to form a three-shot SSDA setting.

**Baselines.** S+T baseline directly trains a model using the labeled source data and labeled target data without unlabeled target data. For the UDA methods (DANN [4], ADR [17], CDAN [11]) as baselines, we modify their training strategies following [18] so that the models can be trained with all the labeled source set, labeled target set and unlabeled target set. ENT [6] is a baseline method using standard entropy minimization for unlabeled target data. MME is the method proposed in [18]. We also design a baseline TML_SPL with a selective pseudo-labeling strategy to compare with our reinforcement learning based selective pseudo-labeling method. TML_SPL uses the target margin loss and selects the most confident pseudo-labeled samples with a threshold of 0.9 to assist to train an improved model.

## 4.2 Implementation Details

We use ResNet-34 [7] for experiments on DomainNet and VGG-16 [19] for experiments on Office-31 and Office-Home, finetuned from ImageNet pre-trained models [15]. We adopt mini-batch SGD with momentum of 0.9 and the learning rate adjusting schedule as [5]. The weight decay is set at 0.0005. As for $r$ and $m$ in Eq. (1), a very small $r$ or a very large $m$ can make the model difficult to converge while a very large $r$ or a very small $m$ can make the margin ignored during training. Therefore, we choose an appropriate pair of values 30 and 0.5 for $r$ and $m$ respectively following previous large margin works [2]. As for the trade-off parameters, we set $\alpha$ in Eq. (3) at 0.1 to keep the two losses at a similar scale so that neither of them will be ignored during the training. $\beta$ and $\lambda$ balance the second and the third terms in Eq. (4). The first two terms in Eq. (4) are both logarithmic forms and the third term is the decrease of the entropy. Thus, we set $\beta$ and $\lambda$ at 1 and 0.1 to keep the three terms at a similar scale so that all the terms can make sense. For the deep Q-learning, we apply the $\varepsilon$-greedy strategy [12] and the experience replay strategy [8]. The $\varepsilon$-greedy strategy is used for the reason that the output by the deep Q-network at early stage cannot reflect the reward and the deep Q-network needs more diverse training samples. The experience replay strategy can make the deep Q-network learn from both current and past information. Our deep Q-network is composed of three fully connected layers, with two hidden layers of 1024 and 512 units respectively. The discount factor $\gamma$ is set to 0.9.

## 4.3 Validation Experiments

In order to show the effectiveness of our method, we design several validation experiments on the 7 scenarios in DomainNet.

**Extending Margin to Source Domain.** We extend the margin parameter to source domain to form a complete margin loss (CML) for comparison, which has a similar formulation to

| Method | R to C | R to P | P to C | C to S | S to P | R to S | P to R | Mean |
|---|---|---|---|---|---|---|---|---|
| CML | 66.3 | 63.8 | 67.2 | 58.8 | 61.0 | 57.2 | 71.7 | 63.7 |
| TML (Ours) | **72.5** | **71.6** | **72.9** | **61.0** | **67.7** | **62.8** | **79.2** | **69.8** |

Table 1: Comparisons with the complete margin loss.

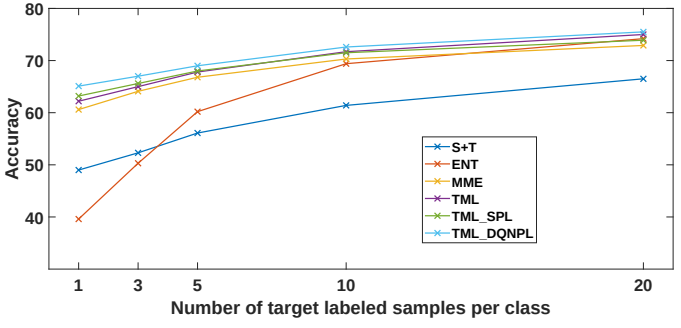


Figure 3: Accuracy vs the number of target labeled samples per class.

the original margin loss and can be written as:

$$\mathcal{L}_{cml} = \frac{1}{N_{s+t}} \sum_{i=1}^{N_{s+t}} \log \frac{e^{r\cos\left(\langle F(\mathbf{x}_i), \mathbf{W}_{y_i}\rangle + m\right)}}{e^{r\cos\left(\langle F(\mathbf{x}_i), \mathbf{W}_{y_i}\rangle + m\right)} + \sum_{j=1, j\neq y_i}^{K} e^{r\cos\langle F(\mathbf{x}_i), \mathbf{W}_j\rangle}} \tag{10}$$

where $s+t$ means labeled source set and labeled target set. The CML method replaces the $\mathcal{L}_{tml}$ in Eq. (3) with $\mathcal{L}_{cml}$. As shown in Table 1, our TML performs much better than CML though the difference between the two methods is quite small. The reason can be that if both the labeled source data and the labeled target data is constrained by the margin then the decision boundary will still be more dependent on the source domain with much more data.

**Varying Number of Target Labeled Samples.** We verify the number of labeled samples in target domain from 1 to 20 per class to explore the performance of our method in different settings. As illustrated in Fig. 3, our TML method can outperform MME and ENT in all the settings while MME gradually performs worse than the simple ENT baseline as the number increasing. Furthermore, when the target labeled samples are much enough, the confidence based selective pseudo-labeling method TML_SPL cannot work well and can even hurt the original model. Our reinforcement learning based selective pseudo-labeling method TML_DQNPL can always make progress to the base model due to the representative and accurate selected pseudo-labels.

## 4.4 Results

The results of our main experiments on the large-scale DomainNet dataset are shown in Table 2. Compared with the UDA methods and the state-of-the-art SSDA method MME [18], our method with only target margin loss (TML) can perform better except for only one case where it performs similarly to MME. On the basis of TML, our final method TML_DQNPL with deep Q-network for selective pseudo-labeling can outperform the baseline TML_SPL

| Method | DomainNet | | | | | | | | Office-31 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | R to C | R to P | P to C | C to S | S to P | R to S | P to R | Mean | W to A | D to A | Mean |
| S+T | 60.0 | 62.2 | 59.4 | 55.0 | 59.5 | 50.1 | 73.9 | 60.0 | 73.2 | 73.3 | 73.3 |
| DANN [9] | 59.8 | 62.8 | 59.6 | 55.4 | 59.9 | 54.9 | 72.2 | 60.7 | 75.4 | 74.6 | 75.0 |
| ADR [23] | 60.7 | 61.9 | 60.7 | 54.4 | 59.9 | 51.1 | 74.2 | 60.4 | 73.3 | 74.1 | 73.7 |
| CDAN [21] | 69.0 | 67.3 | 68.4 | 57.8 | 65.3 | 59.0 | 78.5 | 66.5 | 74.4 | 71.4 | 72.9 |
| ENT [8] | 71.0 | 69.2 | 71.1 | 60.0 | 62.1 | 61.1 | 78.6 | 67.6 | 75.4 | 75.1 | 75.3 |
| MME [23] | 72.2 | 69.7 | 71.7 | 61.8 | 66.8 | 61.9 | 78.5 | 68.9 | 76.3 | 77.6 | 77.0 |
| TML (Ours) | 72.5 | 71.6 | 72.9 | 61.7 | 67.7 | 62.8 | 79.2 | 69.8 | 76.6 | 77.6 | 77.1 |
| TML_SPL | 73.2 | 72.2 | 73.3 | 62.1 | 68.5 | 63.4 | 80.0 | 70.4 | 75.7 | 77.2 | 76.5 |
| TML_DQNPL (Ours) | **75.8** | **74.5** | **75.1** | **64.3** | **69.7** | **64.4** | **82.6** | **72.3** | **77.5** | **78.8** | **78.2** |

Table 2: Results on the 7 scenarios in DomainNet and 2 scenarios in Office-31.

| Method | Office-Home | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R to C | R to P | R to A | P to R | P to C | P to A | A to P | A to C | A to R | C to R | C to A | C to P | Mean |
| S+T | 49.6 | 78.6 | 63.6 | 72.7 | 47.2 | 55.9 | 69.4 | 47.5 | 73.4 | 69.7 | 56.2 | 70.4 | 62.9 |
| DANN | 56.1 | 77.9 | 63.7 | 73.6 | 52.4 | 56.3 | 69.5 | 50.0 | 72.3 | 68.7 | 56.4 | 69.8 | 63.9 |
| ADR | 49.0 | 78.1 | 62.8 | 73.6 | 47.8 | 55.8 | 69.9 | 49.3 | 73.3 | 69.3 | 56.3 | 71.4 | 63.0 |
| CDAN | 50.2 | 80.9 | 62.1 | 70.8 | 45.1 | 50.3 | 74.7 | 46.0 | 71.4 | 65.9 | 52.9 | 71.2 | 61.8 |
| ENT | 48.3 | 81.6 | 65.5 | 76.6 | 46.8 | 56.9 | 73.0 | 44.8 | 75.3 | 72.9 | 59.1 | 77.0 | 64.8 |
| MME | 56.9 | 82.9 | 65.7 | 76.7 | 53.6 | 59.2 | 75.7 | 54.9 | 75.3 | 72.9 | 61.1 | 76.3 | 67.6 |
| TML (Ours) | 56.9 | 83.2 | 67.0 | 76.8 | 54.5 | 59.9 | 75.7 | 54.9 | 75.9 | 73.2 | 61.1 | 77.5 | 68.1 |
| TML_SPL | 55.4 | 82.1 | 67.1 | 76.5 | 55.3 | 60.7 | 75.5 | 53.0 | 75.9 | 73.4 | 60.4 | 77.6 | 67.7 |
| TML_DQNPL (Ours) | **58.4** | **84.0** | **69.1** | **78.5** | **56.8** | **61.7** | **77.0** | **55.9** | **77.1** | **74.5** | **61.9** | **78.8** | **69.5** |

Table 3: Results on the 12 scenarios in Office-Home.

with selective pseudo-labeling by confidence, which demonstrates that our deep Q-network can help select more representative and accurate pseudo-labels.

The results on Office-31 and Office-Home are shown in Table 2 and Table 3 respectively. With these small-scale datasets, our TML method also has better performance than MME in most cases and can perform the same as MME in other cases. In addition, we observe that TML_SPL can hurt the performance in some cases while our TML_DQNPL also makes a progress. The potential reason can be that adding mis-labeled target samples for training causes more damage to the model when the number of the original training samples is small. Therefore, these comparisons can further confirm the effectiveness of our method.

# 5 Conclusions

We propose a novel selective pseudo-labeling method with reinforcement learning for SSDA. We first design a target margin loss for the base model training, which can make the feature distribution closer to the target domain and improve the discriminative ability. Then we apply deep Q-learning to train an agent to select more representative and accurate pseudo-labeled samples for the improved model training. Our method obtains competitive results on several domain adaptation benchmarks and outperforms the present state-of-the-art methods. In addition, the training of the deep Q-network is unrelated to the architecture of base model and will not change the training strategy so that the proposed pseudo-labeling agent can be combined with other advanced methods to help train improved models in the SSDA setting.

# References


[1] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*, 2019.

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[3] Jeff Donahue, Judy Hoffman, Erik Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013.

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[5] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research (JMLR)*, 17(1): 2096–2030, 2016.

[6] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, 2005.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[8] Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3-4):293–321, 1992.

[9] Bingyu Liu, Weihong Deng, Yaoyao Zhong, Mei Wang, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Fair loss: Margin-aware reinforcement learning for deep face recognition. In *ICCV*, 2019.

[10] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[11] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529, 2015.

[13] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[14] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[16] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[17] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Adversarial dropout regularization. In *ICLR*, 2018.

[18] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *ICCV*, 2019.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[20] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.

[21] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018.

[22] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *ACM MM*, 2018.

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

[24] Qian Wang, Penghui Bu, and Toby P Breckon. Unifying unsupervised domain adaptation and zero-shot visual recognition. In *IJCNN*, 2019.

[25] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *CVPR*, 2015.

[26] Serena Yeung, Vignesh Ramanathan, Olga Russakovsky, Liyue Shen, Greg Mori, and Li Fei-Fei. Learning to learn from noisy web videos. In *CVPR*, 2017.

[27] Jianfu Zhang, Naiyan Wang, and Liqing Zhang. Multi-shot pedestrian re-identification via sequential decision making. In *CVPR*, 2018.

[28] Jing Zhang, Wanqing Li, and Philip Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*, 2017.