Unsupervised Domain Adaptive Re-Identification: Theory and Practice

Liangchen Song[1,2]* **Cheng Wang**[2,3]* **Lefei Zhang**[1] **Bo Du**[1]
Qian Zhang[2] **Chang Huang**[2] **Xinggang Wang**[3]
[1]Wuhan University [2]Horizon Robotics [3] Huazhong Univ. of Science and Technology
{wangcheng,xgwang}@hust.edu.cn,{zhanglefei,remoteking}@whu.edu.cn
{liangchen.song,qian01.zhang,chang.huang}@horizon.ai

Abstract

We study the problem of unsupervised domain adaptive re-identification (re-ID) which is an active topic in computer vision but lacks a theoretical foundation. We first extend existing unsupervised domain adaptive classification theories to re-ID tasks. Concretely, we introduce some assumptions on the extracted feature space and then derive several loss functions guided by these assumptions. To optimize them, a novel self-training scheme for unsupervised domain adaptive re-ID tasks is proposed. It iteratively makes guesses for unlabeled target data based on an encoder and trains the encoder based on the guessed labels. Extensive experiments on unsupervised domain adaptive person re-ID and vehicle re-ID tasks with comparisons to the state-of-the-arts confirm the effectiveness of the proposed theories and self-training framework. Our code is available on GitHub.

1 Introduction

To re-identify a particular is to identify it as (numerically) the same particular as one encountered on a previous occasion[24]. Image/video re-identification (re-ID) is a fundamental problem in computer vision and re-ID techniques serve as an indispensable tool for numerous real life applications, for instance, person re-ID for public safety [34], travel time measurement via vehicle re-ID [5]. The key component of re-ID tasks is the notion of identity, which makes re-ID tasks quite different from traditional classification tasks in the following ways: Firstly, determining the label involves two samples, i.e., there is no label when an individual sample is given; secondly, in re-ID tasks the samples in test sets belong to a previously unseen identity while in classification tasks the test samples all fall into a known class. Take the person re-ID task as an example, our target is to spot a person of interest in an image set, which do not have a specific class and is not accessible in the training set.

In many practical situations, we face the problem that the training data and testing data are in different domains. Going back to the person re-ID example, data from a new camera is placed in a new environment, i.e., a new domain is added, which are too costly and impractical to be annotated, a serviceable re-ID model should have a satisfactory accuracy on unlabeled data. Unsupervised domain adaptation means that learning a model for target domain when given both a fully annotated source dataset and an unlabeled target dataset. Existing algorithms for unsupervised domain adaptive re-ID tasks typically learn domain-invariant representation or generate data for target domain through some newly designed networks, which are practical solutions but lack theoretical support [14, 30, 6]. Meanwhile, current theoretical analysis of unsupervised domain adaptation are only concerned with classification tasks [2, 3, 17], which is not suitable for re-ID tasks. A theoretical guarantee of the domain adaptive re-ID problem is in need.

*Equally contribution

Preprint. Work in progress.

In this paper, we first theoretically analyze unsupervised domain adaptive re-ID tasks based on [3], in which three assumptions are introduced for classification. It is assumed that the source domain and the target domain share a same label space in [3]. However, in re-ID tasks, the notion of label is defined for pairwise data and the label indicates a data pair belongs to a same ID or not. We adapt the three assumptions for the input space of pairwise data. Moreover, instead of imposing the assumptions on the raw data as [3], we assume the resemblance between the feature space of two domains. The first assumption is that the criteria of classifying feature pairs is the same between two domains, which is referred to as covariate assumption. The second one is Separately Probabilistic Lipschitzness, indicating that the feature pairs can be divided into clusters. And the last assumption is weight ratio, concerning the probability of existing a repeated feature among all the features from the two domains. Based on the three assumptions, we show the learnability of unsupervised domain adaptive re-ID tasks. Moreover, since our guarantee is built on well extracted features from real images, the encoder, i.e. feature extractor, is trained via a novel self-training framework, which is originally proposed for NLP tasks [19, 20]. Concretely, we iteratively refine the encoder by making guesses on unlabeled target domain and then train the encoder with these samples. In the light of the mentioned assumptions, we propose several loss functions on the encoder and samples with guessed label. And the problem of selecting which sample with guessed label to train is optimized by minimizing the proposed loss functions. For the Separately Probabilistic Lipschitzness assumption, we wish to minimize the intra-cluster and inter-cluster distance. Then the sample selecting problem is turned into data clustering problem and minimizing loss functions is transformed into finding a distance metric for the data. Also, another metric for Weight Ratio is designed. After combining the two metrics together, we have a distance evaluating the confidence of the guessed labels. Finally, the DBSCAN clustering method [5] is employed to generate data clusters according to a threshold on the distance. With pseudo-labels on selected data cluster from target domain, the encoder is trained with triplet loss [32], which is effective for re-ID tasks.

We carry out experiments on diverse re-ID tasks, which demonstrate the priority of our framework. First the well studied person re-ID task is tested and we show the adaptation results between two large scale datasets, i.e. Market-1501 [33] and DukeMTMC-reID [25]. Then we evaluate our algorithm on vehicle re-ID task, in which larger datasets VeRi-776 [16] and PKU-VehicleID [15] are involved. To sum up, the structure of our paper is shown in Figure 2 and our contributions are as follows:

- We introduce the theoretical guarantees of unsupervised domain adaptive re-ID based on [3]. A learnability result is shown under three assumptions that concerning the feature space. To the best of our knowledge, our paper is the first theoretical analysis work on domain adaptive re-ID tasks.
- We theoretically turn the goal of satisfying the assumptions into tractable loss functions on the encoder network and data samples.
- A self-training scheme is proposed to iteratively minimizing the loss functions. Our framework is applicable to all re-ID tasks and the effectiveness is verified on large-scale datasets for diverse re-ID tasks.

1.1 Related work

Unsupervised domain adaptation has been widely studied for decades and the algorithms are divided into four categories in a survey [18]. Using the notions in the survey, our proposed method can be viewed as a combination of feature representation and self-training. Nevertheless, recently unsupervised domain adaptation is widely studied for the person re-ID task.

Unsupervised domain adaptation and feature representation. Feature representation based methods try to find a latent feature space shared between domains. In [26], they wish to minimize a distance between means of the two domains. In a more general manner, [22] and [4] try to find a feature space in which the source and target distributions are similar and the statistic Maximum Mean Discrepancy (MMD) is employed. Also, [10] utilize features that cannot discriminate between source and target domains. Our method and these methods have a same intuition that some features from the source and target domain are generalizable. However, unlike these methods, the process of approximating the intuition in our method is in an iterative manner and we do not directly optimize on the distribution of target domain features.

Unsupervised domain adaptation and self-training. Self-training methods make guesses on target domain and iteratively refine the guesses and are closely related to the Expectation Maximization

(EM) algorithm [21]. In [27], they increase the weight on the target data at each iteration, which is actually altering the relative contribution of source and target domains. A more similar work is [1], in which the model is initially trained on source domain, and then the top-1 recognition hypotheses on the target domain are used for adapting their language model. In our algorithm, we do not guess the labels since different re-ID datasets have totally different labels (identities) and instead we perform clustering on the data.

Unsupervised domain adaptive person re-ID. Due to the rapid development of person re-ID techniques, some useful unsupervised domain adaptive person re-ID methods are proposed. [23] adopt a multi-task dictionary learning scheme to learn a view-invariant representation. Besides, generative models are also applied to domain adaptation in [6, 31]. Wang et al. [30] design a network learning an attribute-semantic and identity discriminative feature representation. Similarly, Li et al. [14] leverages information across datasets and derives domain-invariant features via an adaptation and a re-ID network. Though all the above methods solve the adaptation problem, they are not supported by a theoretical framework and their generalization abilities are not verified in other re-ID tasks. Fan et al. [8] propose a progressive unsupervised learning method consisting of clustering and fine-tuning the network, which is similar to our self-training scheme. However, they only focuses on unsupervised learning, not unsupervised domain adaptation. In addition, their iteration framework is not guided by specific assumptions thus having no theoretical derived loss functions as ours.

2 Notations and Basic Definitions

In classification tasks, let $\mathbf{X} \subseteq \mathbb{R}^d$ be the input space and $\mathbf{Y} \subset \mathbb{R}$ be the output space, and each sample from the input space is denoted by black lower case letters $\mathbf{x} \in \mathbf{X}$. We denote source domain as $\mathcal{S}$ and target domain as $\mathcal{T}$, and both of them are probability distribution over the input space $\mathbf{X}$. Moreover, the real label of each sample is denoted by a labeling function $l : \mathbf{X} \rightarrow \mathbf{Y}$. However, the above notations could not be directly used to analyze the re-ID tasks, because there is no same identity in two domains, i.e. $\mathcal{S}$ and $\mathcal{T}$ do not have the same output (label) space. Fortunately, for re-ID tasks, by treating re-ID as classifying same or different data pairs we are still able to utilize the notations and former results with some simple reformulations.

Specifically, in re-ID tasks, we have a training set consist of data pairs, which means that the input space is $(\mathbf{Z}, \mathbf{Z}) \subseteq \mathbb{R}^{n \times n}$, and the output space is $\mathbf{Y} = \{0, 1\}$, where 1 means the identities in the pair are the same and 0 means different. Observing that in re-ID tasks, the two domain indeed have some overlapping cues, such as color of clothes, wearing a backpack or not in person re-id. That is, we can encode the original data from the two domains with some feature variables or latent variables, and then it is reasonable to assume that distribution of features from two domains satisfy some criteria just as the assumptions used in [3] for classification tasks. Formally, we denote the feature encoder as $\mathbf{x}(\cdot)$ and $\mathbf{x} : \mathbf{Z} \rightarrow \mathbb{R}^d$, and then the labeling function is $l : \mathbf{X} \times \mathbf{X} \rightarrow \{0, 1\}$, where $\mathbf{X} \subseteq \mathbb{R}^d$ is the extracted feature space. For simplicity, we denote $l(\mathbf{x}(\mathbf{z}_1), \mathbf{x}(\mathbf{z}_2)) = l(\mathbf{x}_1, \mathbf{x}_2)$, where $\mathbf{z}_1, \mathbf{z}_2$ means two different raw data. Note that the labeling function is symmetric, i.e. $l(\mathbf{x}_1, \mathbf{x}_2) = l(\mathbf{x}_2, \mathbf{x}_1)$.

3 Assumptions and DA-Learnability

In this section, we first introduce some assumptions reflecting how the source domain interacting with target domain. Then with these assumptions we show the learnability of unsupervised domain adaptive re-ID.

The first assumption is covariate shift, which means that the criteria of classifying data pairs are the same for source domain and target domain. In other words, we have $l_{\mathcal{S}}(\mathbf{x}) = l_{\mathcal{T}}(\mathbf{x})$ for classification tasks, and similarly we can define the covariate shift for re-id tasks on the extracted feature space.

Definition 1 (Covariate Shift)**.** *We say that source and target distribution satisfy the covariate shift assumption if they have the same labeling function, i.e. if we have* $l_{\mathcal{S}}(\mathbf{x}_1, \mathbf{x}_2) = l_{\mathcal{T}}(\mathbf{x}_1, \mathbf{x}_2)$*.*

Another assumption is inspired by the "Probabilistic Lipschitzness", which is originally proposed for semi-supervised learning in [28] and then investigated with application to domain adaptation tasks in [3]. This assumption captures the intuition that in a classification task, the data can be divided into label-homogeneous clusters and are separated by low-density regions. However, in re-id tasks, the labeling function is a multivariable function, thus the original Probabilistic Lipschitzness is not applicable. Note that the intuition of re-id tasks is that similar pairs can form as a cluster. That is, for an instance, the similar data can be divided into a cluster and the cluster is separated out from the data space with a low-density gap. Mathematically, we have the following definition.

Definition 2 (Separately Probabilistic Lipschitzness (SPL)). *Let $\phi : \mathbb{R} \to [0,1]$ be monotonically increasing. Symmetric function $f : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ is ϕ-SPL with respect to a distribution $\mathcal{D}$ on $\mathbf{X}$, if for all $\lambda > 0$,*

$$\underset{\mathbf{x}_1,\mathbf{x}_2 \sim \mathcal{D}}{\mathbb{P}} \left(\exists \mathbf{y} : |f(\mathbf{x}_1,\mathbf{x}_2) - f(\mathbf{x}_1,\mathbf{y})| \geqslant \lambda \|\mathbf{x}_2 - \mathbf{y}\|\right) \leqslant \phi(\lambda) \tag{1}$$

To ensure the learnability of the domain adaptation task, we still need a critical assumption concerning how much overlap there is between the source and target domain. We again follow the assumption used in [3] on the source and target distribution, which is a relaxation of the pointwise density ratio between the two distributions.

Definition 3 (Weight Ratio). *Let $\mathfrak{B} \subseteq 2^{\mathbf{X}}$ be a collection of subsets of the input space $\mathbf{X}$ measurable with respect to both $\mathcal{S}$ and $\mathcal{T}$. For some $\eta > 0$ we define the η-weight ratio of the source distribution and the target distribution with respect to $\mathfrak{B}$ as*

$$C_{\mathfrak{B},\eta}(\mathcal{S},\mathcal{T}) = \inf_{\substack{b \in \mathfrak{B} \\ \mathcal{T}(b) \geqslant \eta}} \frac{\mathcal{S}(b)}{\mathcal{T}(b)} \tag{2}$$

Further, we define the weight ratio of the source distribution and the target distribution with respect to $\mathfrak{B}$ as

$$C_{\mathfrak{B}}(\mathcal{S},\mathcal{T}) = \inf_{\substack{b \in \mathfrak{B} \\ \mathcal{T}(b) \neq 0}} \frac{\mathcal{S}(b)}{\mathcal{T}(b)} \tag{3}$$

Following the notations in [3], we also assume that our domain is the unit cube $\mathbf{X} = [0,1]^d$ and let $\mathfrak{B}$ denote the set of axis aligned rectangles in $[0,1]^d$. For our re-ID tasks, the risk of a classifier h on target domain is

$$\mathrm{R}_{\mathcal{T}}(h) = \underset{\mathbf{x}_1,\mathbf{x}_2 \sim \mathcal{T}}{\mathbb{E}} [\mathcal{L}(h(\mathbf{x}_1,\mathbf{x}_2), l(\mathbf{x}_1,\mathbf{x}_2))]. \tag{4}$$

Let the Nearest Neighbor classifier be h_{NN}, then the following theorem implies the learnability of domain adaptive re-ID, of which the proof is included in supplemental materials.

Theorem 1. *Let the domain be the unit cube,$\mathbf{X} = [0,1]^d$, and for some $C > 0$, let $(\mathcal{S},\mathcal{T})$ be a pair of source and target distributions over $\mathbf{X}$ satisfying the covariate shift assumption, with $C_{\mathfrak{B}}(\mathcal{S},\mathcal{T}) \geqslant C$, and their common deterministic labeling function $l : \mathbf{X} \times \mathbf{X} \to \{0,1\}$ satisfying the ϕ-SPL property with respect to the target distribution, for some function ϕ. Then, for all $\epsilon, \delta > 0$, for all $(\mathcal{S},\mathcal{T})$, if S is a source generated sample set of size at least*

$$m \geqslant \frac{4}{\epsilon \delta C e} \left(\phi^{-1}\left(\frac{\epsilon}{4}\right)\sqrt{d}\right)^d$$

then, with probability at least $1-\delta$ (over the choice of S), $\mathrm{R}_{\mathcal{T}}(h_{\mathrm{NN}})$ is at most ϵ.

4 Reinforcing the Assumptions

In previous section, we show that with some assumptions on the extracted feature space, unsupervised domain adaptation is learnable. Thus we are concerned with how to train a feature extractor, i.e. encoder, satisfying the mentioned assumptions. Briefly speaking, we first derive several loss functions according to the assumptions and then iteratively train the encoder to minimize the loss functions via a self-training framework.

Self-training framework. Assume that we have an encoder $\mathbf{x}$ and some samples $\mathcal{D}$ with guessed label l on target domain, and the loss function is $\mathcal{L}(\mathbf{x},\mathcal{D},l)$. In self-training, at first a $\mathbf{x}^{(i)}$ is used to extract features from all available unlabeled samples, and the target now is minimizing the loss through selecting samples, that is $\min_{\mathcal{D},l} \mathcal{L}(\mathbf{x}^{(i)},\mathcal{D},l)$. On the next round, with these selected samples, the encoder $\mathbf{x}^{(i)}$ is updated by solving the minimization problem $\min_{\mathbf{x}} \mathcal{L}(\mathbf{x},\mathcal{D}^{(i)},l^{(i)})$.

It is worthwhile to note that the covariate shift assumption only depends on the property of labeling function, thus in this section we only consider the proposed SPL and weight ratio.

4.1 Reinforcing the SPL

Recall that the original data is $\mathbf{z} \in \mathbf{Z}$ and we wish to iteratively find a encoder $\mathbf{x}(\cdot)$ such that in the feature space the SPL property is satisfied as much as possible. So we first need a definition to evaluate whether one encoder is better than another concerning the SPL property.

Definition 4. *Encoder $\mathbf{x}^a(\cdot)$ is said to be more clusterable than $\mathbf{x}^b(\cdot)$ with respect to a labeling function l and a distribution $\mathcal{D}$ over $\mathbf{Z}$, if there exists $\epsilon \in (0,1)$, and $\lambda \in \{\lambda_1, \lambda_2\}$ with $\lambda_1\lambda_2 < 0$, such that*

$$\begin{aligned}&\mathop{\mathbb{P}}_{\mathbf{z}_1,\mathbf{z}_2\sim\mathcal{D}}\big(\exists\mathbf{z}_3 : |l(\mathbf{x}^a(\mathbf{z}_1), \mathbf{x}^a(\mathbf{z}_2)) - l(\mathbf{x}^a(\mathbf{z}_1), \mathbf{x}^a(\mathbf{z}_3))| - \epsilon \geqslant \lambda\|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\|\big)\\ &\qquad\leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1,\mathbf{z}_2\sim\mathcal{D}}\big(\exists\mathbf{z}_3 : |l(\mathbf{x}^b(\mathbf{z}_1), \mathbf{x}^b(\mathbf{z}_2)) - l(\mathbf{x}^b(\mathbf{z}_1), \mathbf{x}^b(\mathbf{z}_3))| - \epsilon \geqslant \lambda\|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\|\big)\end{aligned}$$

The above equation differs from the original SPL (1) for the reason that the original form is too strict to be satisfied. Now we can easily define a loss function

$$\mathcal{L}(\mathbf{x}, \mathcal{D}, l; \epsilon, \lambda) = \mathop{\mathbb{E}}_{\mathbf{z}_1,\mathbf{z}_2\sim\mathcal{D}}\big[\exists\mathbf{z}_3 : |l(\mathbf{x}(\mathbf{z}_1), \mathbf{x}(\mathbf{z}_2)) - l(\mathbf{x}(\mathbf{z}_1), \mathbf{x}(\mathbf{z}_3))| - \epsilon \geqslant \lambda\|\mathbf{x}(\mathbf{z}_2) - \mathbf{x}(\mathbf{z}_3)\|\big],$$

where $\mathcal{D}$ means a set of samples and l is the guessed labeling function. However, directly performing optimization on the loss function is infeasible since the analytical form is unknown. To overcome the difficulty, we adopt intra-cluster distance and inter-cluster distance,

$$\mathcal{L}_{\text{intra}}(\mathbf{x}, \mathcal{D}, l) = \sum_{l(\mathbf{x}(\mathbf{z}_1),\mathbf{x}(\mathbf{z}_2))=1} \|\mathbf{x}(\mathbf{z}_1) - \mathbf{x}(\mathbf{z}_2)\|, \tag{5}$$

$$\mathcal{L}_{\text{inter}}(\mathbf{x}, \mathcal{D}, l) = \sum_{l(\mathbf{x}(\mathbf{z}_1),\mathbf{x}(\mathbf{z}_2))=0} -\|\mathbf{x}(\mathbf{z}_1) - \mathbf{x}(\mathbf{z}_2)\|. \tag{6}$$

We show that minimizing $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$ is appropriate for being more clusterable through the following theorem.

Theorem 2. *For two encoders $\mathbf{x}^a, \mathbf{x}^b$, a distribution $\mathcal{D}$ and a labeling function l, then*

$$\mathbf{x}^a \textit{ is more clusterable than } \mathbf{x}^b \Leftrightarrow \begin{cases}\mathcal{L}_{\text{intra}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{intra}}(\mathbf{x}^b, \mathcal{D}, l)\\ \mathcal{L}_{\text{inter}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{inter}}(\mathbf{x}^b, \mathcal{D}, l)\end{cases}$$

For proof we refer reader to the supplemental materials. Here, Definition 4 and Theorem 2 describe how to evaluate an encoder with a fixed distribution $\mathcal{D}$ and labeling function l. Obviously, we can fix the encoder and rewrite the results to evaluate the samples with guessed labels. For the sake of conciseness, the details are omitted. When $\mathcal{D}$ and l are fixed during the iteration procedure, minimizing $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$ are straightforward. Contrastingly, we have to focus more on the strategy of picking out samples with guessed labels.

Selecting samples via clustering. In spite of the similarity between $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$, they do not share a same strategy regarding the sample selection step. For $\mathcal{L}_{\text{intra}}$, if all the data in $\mathcal{T}$ are encoded with a $\mathbf{x}$, then for each pair $(\mathbf{x}_i, \mathbf{x}_j)$, it is natural to assume that a smaller $\|\mathbf{x}_i - \mathbf{x}_j\|$ implies a higher confidence that $l(\mathbf{x}_i, \mathbf{x}_j) = 1$. Likewise, a larger $\|\mathbf{x}_i - \mathbf{x}_j\|$ implies a higher confidence that $l(\mathbf{x}_i, \mathbf{x}_j) = 0$. But choosing a high confidence different pair as training data does not really improve the real performance, because the accuracy is more sensitive about the minimal distance of different pairs, i.e., $\inf_{i:l(\mathbf{x}_d,\mathbf{x}_i)=0} \|\mathbf{x}(\mathbf{z}_d) - \mathbf{x}(\mathbf{z}_i)\|$. So rather than directly selecting different pairs, we treat the selected samples as a series of clusters and dissimilar pairs are selected on the basis of different clusters. That is to say, in order to minimize $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$ simultaneously, we perform clustering on the data with guessed labels.

Distance metrics and loss functions. Up to this point, we are facing an unsupervised clustering problem, which is largely settled by the distance metric. In other words, designing a sample selecting strategy to minimize a loss turns into designing a distance metric between samples, and a better distance should lead to a lower $\mathcal{L}_{\text{intra}}$ and $\mathcal{L}_{\text{inter}}$. It is a common practice in image retrieval that the contextual similarity [12] measure is more robust and beneficial for a lower $\mathcal{L}_{\text{intra}}$.

In our practice, we adopt the k-reciprocal encoding in [35] as the distance metric, which is a variation of Jaccard distance between nearest neighbors sets. Precisely, with an encoder $\mathbf{x}$, all samples from $\mathcal{T}$ are encoded and with these features a distance matrix $M \in \mathbb{R}^{m_t\times m_t}$ is computed where $M_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ and m_t is the total number of target samples. Then M is updated by

$$M_{ij} = \begin{cases} e^{-M_{ij}} & \text{if } j \in \mathcal{I}_i,\\ 0 & \text{otherwise.}\end{cases} \tag{7}$$

where the indices set $\mathcal{I}_i$ is the so called robust set for $\mathbf{x}_i$. $\mathcal{I}_i$ is determined by first choosing mutual k nearest neighbors for the probe, then incrementally adding elements. Specifically, denote the indices of mutual k nearest neighbors of $\mathbf{x}_i$ as $\mathcal{K}_k(\mathbf{x}_i)$ and then for all $s \in \mathcal{K}_k(\mathbf{x}_i)$, if $|\mathcal{K}_k(\mathbf{x}_i) \cap \mathcal{K}_{\frac{k}{2}}(\mathbf{x}_s)| \geqslant \frac{2}{3}|\mathcal{K}_{\frac{k}{2}}(\mathbf{x}_s)|$, let $\mathcal{I}_i \leftarrow \mathcal{K}_k(\mathbf{x}_i) \cup \mathcal{K}_{\frac{k}{2}}(\mathbf{x}_s)$. In particular, for a pair $(\mathbf{x}_i, \mathbf{x}_j)$, we have

$$d_{\mathrm{J}}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\sum_{k=1}^{m_t} \min(M_{ik}, M_{jk})}{\sum_{k=1}^{m_t} \max(M_{ik}, M_{jk})}. \tag{8}$$

4.2 Reinforcing the weight ratio

As mentioned before, weight ratio is a crucial part to support the learnability of domain adaptation. Apart from directly define a loss based on the original weight ratio definition, a similar way as the SPL case is minimizing the loss

$$\mathcal{L}_{\mathrm{WR}}(\mathbf{x}, \mathcal{D}) = \mathop{\mathbb{E}}_{\mathbf{z}_d \sim \mathcal{D}} \left[\inf_{\mathbf{z}_s \sim \mathcal{S}} \|\mathbf{x}(\mathbf{z}_d) - \mathbf{x}(\mathbf{z}_s)\| \right], \tag{9}$$

where $\mathcal{S}$ is the source domain. The intuition here is to enhance the degree of similarity, which means that each target feature is close to some source features. We denote $C_{\mathfrak{B},\eta}(\mathcal{S}, \mathcal{T}; \mathbf{x})$ as the weight ratio when using $\mathbf{x}$ as the encoder, where $\mathfrak{B}$ is defined in Section 3. The following theorem demonstrate that our $\mathcal{L}_{\mathrm{WR}}$ makes sense and the proof is in the supplemental materials.

Theorem 3. *For two encoders $\mathbf{x}^a, \mathbf{x}^b$, a distribution $\mathcal{D}$, if η is a random variable and its support is a subset of $\mathbb{R}^+$, then*

$$\mathcal{L}_{\mathrm{WR}}(\mathbf{x}^a, \mathcal{D}) \leqslant \mathcal{L}_{\mathrm{WR}}(\mathbf{x}^b, \mathcal{D}) \Leftrightarrow \mathbb{E}\left[C_{\mathfrak{B},\eta}(\mathcal{S}, \mathcal{D}; \mathbf{x}^a)\right] \geqslant \mathbb{E}\left[C_{\mathfrak{B},\eta}(\mathcal{S}, \mathcal{D}; \mathbf{x}^b)\right]$$

However, unlike $\mathcal{L}_{\mathrm{inter}}$ and $\mathcal{L}_{\mathrm{intra}}$, it is hard to optimize on $\mathbf{x}$ for $\mathcal{L}_{\mathrm{WR}}$ because of the infimum. On the other hand, selecting samples is easily done via giving more confidence to the sample with smaller $\inf_{\mathbf{z}_s \sim \mathcal{S}} \|\mathbf{x}(\mathbf{z}_d) - \mathbf{x}(\mathbf{z}_s)\|$. More specifically, for each $\mathbf{x}_i$ from $\mathcal{T}$, we search the nearest neighbor in $\mathcal{S}$. The function measuring the confidence for each $\mathbf{x}_i$ is denoted by

$$d_{\mathrm{W}}(\mathbf{x}_i) = 1 - e^{-\|\mathbf{x}_i - N_{\mathcal{S}}(\mathbf{x}_i)\|^2}. \tag{10}$$

where $N_{\mathcal{S}}(\mathbf{x}_i)$ means the nearest neighbor of $\mathbf{x}_i$ in source domain $\mathcal{S}$, and a smaller d_{W} means a higher confidence. To transform d_{W} and d_{J} onto the same scale, we perform a simple normalization on d_{W}, i.e., divided by $\max_i d_{\mathrm{W}}(\mathbf{x}_i)$. Combining with d_{J}, the final distance matrix is $M_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$ and

$$d(\mathbf{x}_i, \mathbf{x}_j) = (1 - \lambda) d_{\mathrm{J}}(\mathbf{x}_i, \mathbf{x}_j) + \lambda(d_{\mathrm{W}}(\mathbf{x}_i) + d_{\mathrm{W}}(\mathbf{x}_j)), \tag{11}$$

where $\lambda \in [0, 1]$ is a balancing parameter.

4.3 Overall algorithm

So far, general outlines of reinforcing the assumptions have been elaborated, except the details about the clustering method. In our framework, a good clustering method should possess the following properties: (a) it does not require the number of clusters as an input, because in fact a cluster means an identity and the number of identities is trivial and unknown; (b) it is able to avoid pairs of low confidence, that is allowing some points not belonging to any clusters; (c) it is scalable enough to incorporate our theoretically derived distance metric. We employ the clustering method named DBSCAN [7], which has stood the test of time and exactly have the mentioned advantages.

Now we provide some other practical details of our domain adaptive re-ID algorithm. At the beginning, an encoder $\mathbf{x}^{(0)}$ is well trained on $\mathcal{S}$ and all the pairs are computed with Eqn.(11). Next, we describe how we set the threshold controlling whether a pair should be used to train. Intuitively, the threshold should be irrelevant to tasks since the scale of d varies from tasks. So in our method, we first sort all the distance from lowest to highest and the average value of top pN pairs is set to be the threshold τ, where N is the total number of possible pairs and p is percentage. On these data with pseudo-labels, the encoder is then trained with triplet loss [32]. Our whole framework is concluded in Algorithm 1.

5 Experiments

In this section, we test our unsupervised domain adaptation algorithm on person re-ID and vehicle re-ID. The performance are evaluated by cumulative matching characteristic (CMC) and mean Average Precision (mAP), which are multi-gallery-shot evaluation metrics defined in [33].

Algorithm 1: Unsupervised Domain Adaptation for Re-ID

```
input  : source domain dataset $S$, unlabeled target domain dataset $T$ with $m_t$ samples, balancing parameter $\lambda$,
         percentage $p$, the minimum size of a cluster $N_1$, iteration number $N_2$
output : an encoder $\mathbf{x}$ for target domain
Train an encoder $\mathbf{x}^{(0)}$ on $S$;
Compute $T^{(0)} = \mathbf{x}^{(0)}(T), S^{(0)} = \mathbf{x}^{(0)}(S)$;
Compute a distance matrix $M^{(0)}$ on $T^{(0)}, S^{(0)}$ by Eqn.(11);
Sort all the $N$ elements in $M^{(0)}$ from low to high and record the mean of top $pN$ values as threshold $\tau$;
Select train data $D^{(0)} = \texttt{DBSCAN}(M^{(0)}; \tau, N_1)$;
Train $\mathbf{x}^{(1)}$ on $D$;
for $i = 1$ to $N_2$ do
    Compute $T^{(i)} = \mathbf{x}^{(i)}(T^{(i-1)}), S^{(i)} = \mathbf{x}^{(i)}(S^{(i-1)})$;
    Compute $M^{(i)}$ on $T^{(i)}, S^{(i)}$;
    Select $D^{(i)} = \texttt{DBSCAN}(M^{(i)}; \tau, N_1)$;
    Train $\mathbf{x}^{(i+1)}$ on $D^{(i)}$;
end
```

Parameter settings and implementation details. In all the following re-ID experiments, we empirically set $\lambda = 0.1, p = 1.6 \times 10^{-3}$, $N_1 = 4$ and $N_2 = 20$. Basically, the encoder is ResNet-50 [11] pre-trained on ImageNet. Both triplet and softmax loss are used for initializing the network on source domain, while only triplet loss is used for refining the encoder on target domain. More details about the network, training parameters and visual examples from different domains are included in the supplemental materials. Moreover, in the supplemental materials we also investigate other distance metrics and clustering methods.

5.1 Person re-ID

Market-1501 [33] and DukeMTMC-reID [25] are two large scale datasets and frequently used for unsupervised domain adaptation experiments. Both of the two datasets are split into a training set and a testing set. The details including the number of identities and images are shown in Table 1.

Table 1: The details of datasets used in our experiments.

Datasets	Training		Testing	
	#IDs	#Images	#IDs	#Images
Market [33]	751	12,936	750	19,732
Duke [25]	702	16,522	702	19,889
VeRi [16]	576	37,778	200	13,257
PKU [15]	2,290	24,157	-	-

Comparison methods are selected in three aspects. Firstly, we show the performance of direct transfer, that is directly using the initial source-trained encoder on the target domain. Also, the plain self-training scheme is compared as a baseline, which means sample selection only depends on their Euclidean distance. Secondly, our method is compared with three most recent state-of-the-art methods[2]: SPGAN [6], TJ-AIDL [30] and ARN [14]. We report the original results quoted from in their papers. Thirdly, we show the results of our methods with and without d_{W}, which can be viewed as ablation studies. The results are shown in Table 2, from which we can observe the following facts: (a) The accuracy of self-training baseline is high and even better than two recent methods, indicating that our clustering based self-training scheme is fairly good; (b) The version without d_{W} is better than self-training baseline, which shows the effectiveness of d_{J}, and after incorporated with d_{W} the final method achieves the highest accuracy, reflecting the advantage of d_{W}. Thus our two assumptions are both useful according to the ablation studies. (c) Although the proposed d_{W} is beneficial, the increase of accuracy brought by it varies from different tasks. We think this is related to the distribution of source and target domains. Please refer to for more discussion in B.3 on this problem.

Furthermore, we draw the mAP curves (Figure 1) during the iterations of the adaptation task Duke→Market, in which self-training baseline, using distance without d_{W} and $\lambda = \{0.05, 0.1, 0.5, 0.7\}$ are compared. We can see that except the baseline, all the curves have a similar

[2] Our results also outperforms PTGAN[31] by large margin, but the comparison with PTGAN is not shown here since we adopt a different backbone network.

Table 2: Comparison of unsupervised domain adaptive person re-ID methods.

Methods	DukeMTMC-reID→Market-1501				Market-1501→DukeMTMC-reID			
	rank-1	rank-5	rank-10	mAP	rank-1	rank-5	rank-10	mAP
Direct Transfer	46.8	64.6	71.5	19.1	27.3	41.2	47.1	11.9
Self-training Baseline	66.7	80.0	85.0	39.6	40.8	53.9	60.5	24.7
SPGAN [6]	57.7	75.8	82.4	26.7	46.4	62.3	68.0	26.2
TJ-AIDL [30]	58.2	74.8	81.1	26.5	44.3	59.6	65.0	23.0
ARN [14]	70.3	80.4	86.3	39.4	60.2	73.9	79.5	33.4
Ours w/o d_{W}	75.1	88.7	92.4	52.5	68.1	**80.1**	83.2	**49.0**
Ours	**75.8**	**89.5**	**93.2**	**53.7**	**68.4**	**80.1**	**83.5**	**49.0**

Table 3: Comparison of unsupervised domain adaptive vehicle re-ID methods.

Methods	PKU-VehicleID→VeRi-776			
	rank-1	rank-5	rank-10	mAP
Direct Transfer	52.1	65.1	71.1	14.6
Self-training Baseline	74.4	81.6	84.6	33.5
SPGAN [6]	57.4	70.0	75.6	16.4
Ours w/o d_{W}	76.7	85.5	**89.3**	35.3
Ours	**76.9**	**85.8**	89.0	**35.8**

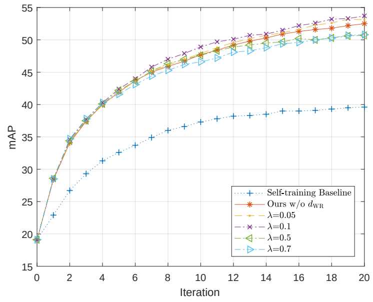


Figure 1: Convergence comparison

tendency toward convergence. A subtle distinction is that after 18 iterations methods with smaller λ become unstable, while methods with larger λ move toward convergence.

5.2 Vehicle re-ID

We use VeRi-776 [16] and part of PKU-VehicleID [15] for vehicle re-ID experiments[3], the details are included in Table 1. Unlike person re-ID, currently there are no unsupervised domain adaptation algorithms designed for vehicle re-ID. Thus, we use the existing solutions for person re-ID as comparisons[4]. As shown in Table 3, not only are the conclusions from person re-ID verified again, but also the generalization ability of our method is shown. We discover that the compared SPGAN generates quite presentable images and we put the images into supplemental materials, but their accuracy is still lower than the self-training baseline, not to mention our proposed method.

6 Conclusion and Future Work

In this work, we bridge the gap between theories of unsupervised domain adaptation and re-id tasks. Inspired by previous work [3], we make assumptions on the extracted feature space and then show the learnability of unsupervised domain adaptive re-id tasks. Treating the assumptions as the goal of our encoder, several loss functions are proposed and then minimized via self-training framework.

Though the proposed solution is effective and outperforms state-of-the-art methods, there are still problems unsolved in our algorithm. Firstly, with regard of the weight ratio assumption, we propose the loss function $\mathcal{L}_{\mathrm{WR}}$, which is ignored when updating the encoder because of the intractable infimum. So designing another feasible loss function is an interesting direction of research. Another promising issue is to improve the data selecting step in the self-training scheme. We turn the data selecting step into a clustering problem, which can be thought of as a version with hard threshold. This suggest that there may be a better strategy which utilize the relative values between distances. We hope that our analyses could open the door to develop new domain adaptive re-ID tasks and can lift the burden of designing large and complicate networks.

[3] In PKU-VehicleID, the camera information is not provided but needed when computing the CMC and mAP, so we only test with the setting that PKU-VehicleID as source dataset and VeRi-776 as target dataset.

[4] We only test SPGAN. Because (1) source code of ARN is not available; (2) TJ-AIDL requires attribute labels as an input, which is not available in vehicle re-ID datasets. For SPGAN, the experiments are carried out with their default parameters for person re-ID.

References

[1] Michiel Bacchiani and Brian Roark. Unsupervised language model adaptation. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 1, pages I–I. IEEE, 2003.

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

[3] Shai Ben-David and Ruth Urner. Domain adaptation–can quantity compensate for quality? *Annals of Mathematics and Artificial Intelligence*, 70(3):185–202, Mar 2014.

[4] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 179–188. ACM, 2009.

[5] Benjamin Coifman. Vehicle re-identification and travel time measurement in real-time on freeways using existing loop detector infrastructure. *Transportation Research Record: Journal of the Transportation Research Board*, (1643):181–191, 1998.

[6] Weijian Deng, Liang Zheng, Qixiang Ye, Guoliang Kang, Yi Yang, and Jianbin Jiao. Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

[8] Hehe Fan, Liang Zheng, and Yi Yang. Unsupervised Person Re-identification: Clustering and Fine-tuning. *arXiv*, May 2017.

[9] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[10] 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*, 17(1):2096–2030, 2016.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[12] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv*, Dec 2014.

[14] Yu-Jhe Li, Fu-En Yang, Yen-Cheng Liu, Yu-Ying Yeh, Xiaofei Du, and Yu-Chiang Frank Wang. Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[15] Hongye Liu, Yonghong Tian, Yaowei Wang, Lu Pang, and Tiejun Huang. Deep relative distance learning: Tell the difference between similar vehicles. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2167–2175, 2016.

[16] X. Liu, W. Liu, H. Ma, and H. Fu. Large-scale vehicle re-identification in urban surveillance videos. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2016.

[17] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain Adaptation: Learning Bounds and Algorithms. *arXiv*, Feb 2009.

[18] Anna Margolis. A literature review of domain adaptation with unlabeled data. *Technical report*, pages 1–42, 2011.

[19] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics, 2006.

[20] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics, 2006.

[21] Kamal Nigam, Andrew McCallum, and Tom Mitchell. Semi-supervised text classification using em. *Semi-Supervised Learning*, pages 33–56, 2006.

[22] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008.

[23] Peixi Peng, Tao Xiang, Yaowei Wang, Massimiliano Pontil, Shaogang Gong, Tiejun Huang, and Yonghong Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1306–1315, 2016.

[24] Alvin Plantinga. Things and persons. *The Review of Metaphysics*, pages 493–519, 1961.

[25] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision workshop on Benchmarking Multi-Target Tracking*, 2016.

[26] Sandeepkumar Satpal and Sunita Sarawagi. Domain adaptation of conditional probability models via feature subsetting. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 224–235. Springer, 2007.

[27] Songbo Tan, Xueqi Cheng, Yuefen Wang, and Hongbo Xu. Adapting naive bayes to domain adaptation for sentiment analysis. In *European Conference on Information Retrieval*, pages 337–349. Springer, 2009.

[28] Ruth Urner, Shai Ben-David, and Shai Shalev-Shwartz. Access to unlabeled data can speed up prediction time. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pages 641–648, USA, 2011. Omnipress.

[29] Nam Vo and James Hays. Generalization in Metric Learning: Should the Embedding Layer be the Embedding Layer? *arXiv*, Mar 2018.

[30] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[31] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] Kilian Q. Weinberger and Lawrence K. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[33] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable Person Re-identification: A Benchmark. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1116–1124, Dec 2015.

[34] Liang Zheng, Yi Yang, and Alexander G. Hauptmann. Person Re-identification: Past, Present and Future. *arXiv*, Oct 2016.

[35] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3652–3661. IEEE, 2017.

Supplementary Materia

A Theorems and Proofs

To prove Theorem 1, we first give a lemma on the upper error bound of $\mathrm{R}_{\mathcal{T}}(h_{\mathrm{NN}})$. Let $\mathfrak{B}$ denote the set of axis aligned rectangles in $[0,1]^d$ and, given some $\eta > 0$, let $\mathfrak{B}_\eta$ denotes the class of axis aligned rectangles with side-length η. For a sample set S from source domain, we have

Lemma 1. *Let the domain be the unit cube, $\mathbf{X} = [0,1]^d$, and for some $C > 0$ and some $\eta \geqslant 0$, let $(\mathcal{S}, \mathcal{T})$ be source and target distributions over $\mathbf{X}$ satisfying the covariate shift assumption, with $C_{\mathfrak{B},\eta}(\mathcal{S}, \mathcal{T}) \geqslant C$, and their common re-id labeling function $l : \mathbf{X} \times \mathbf{X} \to \{0,1\}$ satisfying the ϕ-SPL property with respect to the target distribution, for some function ϕ. Then, for all m, and all $(\mathcal{S}, \mathcal{T})$,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{S}^m}[\mathrm{R}_{\mathcal{T}}(h_{\mathrm{NN}})] \leqslant 2\phi(\frac{1}{\eta\sqrt{d}}) + \frac{2}{\eta^d Cme} \tag{12}$$

Proof. A test pair $(\mathbf{x}_1, \mathbf{x}_2)$ gets the wrong label under two conditions: (a) at least one test data do not have a close neighbor with all the m training data; (b) $(\mathbf{x}_1, \mathbf{x}_2)$ have a close neighbor pair which have the opposite label. For (a), we can use the results from Lemma 7 and Theorem 8 in [3]. Specifically, let $C_1, C_2, \cdots, C_1/\eta^d$ be a cover of the set $[0,1]^d$ using boxes of side-length η. We have

$$\mathop{\mathbb{E}}_{S\sim\mathcal{S}^m}\left[\sum_{i:S\cap C_i=\varnothing}\mathcal{T}(C_i)\right] \leqslant \frac{1}{\eta^d Cme}. \tag{13}$$

If $\mathbf{x}$ is in the box $C_{\mathbf{x}}$, then the probability of (a) can be expressed as $\mathbb{P}(C_{\mathbf{x}_1} \cap S = \varnothing \vee C_{\mathbf{x}_2} \cap S = \varnothing)$. Observing that

$$\mathbb{P}(C_{\mathbf{x}_1} \cap S = \varnothing \vee C_{\mathbf{x}_2} \cap S = \varnothing) \leqslant \mathbb{P}(C_{\mathbf{x}_1} \cap S = \varnothing) + \mathbb{P}(C_{\mathbf{x}_2} \cap S = \varnothing)$$

and $\mathbb{P}(C_{\mathbf{x}} \cap S = \varnothing) = \sum_{i:S\cap C_i=\varnothing}\mathbb{P}(C_i)$, so (a) is bounded by $\frac{2}{\eta^d Cme}$. For (b), we denote the nearest neighbor to $\mathbf{x}$ in S is $N_S(\mathbf{x})$ and then (b) means in the box we have

$$l(\mathbf{x}_1, \mathbf{x}_2) \neq l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2)) \wedge \|N_S(\mathbf{x}_1) - \mathbf{x}_1\| \leqslant \eta\sqrt{d} \wedge \|N_S(\mathbf{x}_2) - \mathbf{x}_2\| \leqslant \eta\sqrt{d}. \tag{14}$$

Seeing that

$$\begin{aligned}
&|l(\mathbf{x}_1, \mathbf{x}_2) - l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2))| \\
&\qquad = |l(\mathbf{x}_1, \mathbf{x}_2) - l(\mathbf{x}_1, N_S(\mathbf{x}_2)) + l(\mathbf{x}_1, N_S(\mathbf{x}_2)) - l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2))| \\
&\qquad \leqslant |l(\mathbf{x}_1, \mathbf{x}_2) - l(\mathbf{x}_1, N_S(\mathbf{x}_2))| + |l(\mathbf{x}_1, N_S(\mathbf{x}_2)) - l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2))|
\end{aligned}$$

So

$$\begin{aligned}
\mathbb{P}\Big(l(\mathbf{x}_1, \mathbf{x}_2) \neq & l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2)) \wedge \|N_S(\mathbf{x}_1) - \mathbf{x}_1\| \leqslant \eta\sqrt{d} \wedge \|N_S(\mathbf{x}_2) - \mathbf{x}_2\| \leqslant \eta\sqrt{d}\Big) \\
&\leqslant \mathbb{P}\left(|l(\mathbf{x}_1, \mathbf{x}_2) - l(\mathbf{x}_1, N_S(\mathbf{x}_2))| \geqslant \frac{1}{\eta\sqrt{d}}\|N_S(\mathbf{x}_2) - \mathbf{x}_2\|\right) \\
&\qquad + \mathbb{P}\left(|l(\mathbf{x}_1, N_S(\mathbf{x}_2)) - l(N_S(\mathbf{x}_1), N_S(\mathbf{x}_2))| \geqslant \frac{1}{\eta\sqrt{d}}\|N_S(\mathbf{x}_1) - \mathbf{x}_1\|\right) \\
&\leqslant 2\phi(\frac{1}{\eta\sqrt{d}})
\end{aligned}$$

Combining the two bounds together, we conclude our proof. □

If we have a stronger weight ratio assumption, i.e. $C_{\mathfrak{B}}(\mathcal{S}, \mathcal{T}) \geqslant C$, we get the following result of domain adaptation learnability.

Theorem 4. *Let the domain be the unit cube,$\mathbf{X} = [0,1]^d$, and for some $C > 0$, let $(\mathcal{S}, \mathcal{T})$ be a pair of source and target distributions over $\mathbf{X}$ satisfying the covariate shift assumption, with $C_{\mathfrak{B}}(\mathcal{S}, \mathcal{T}) \geqslant C$, and their common deterministic labeling function $l : \mathbf{X} \times \mathbf{X} \to \{0,1\}$ satisfying the ϕ-SPL property* *with respect to the target distribution, for some function ϕ. Then, for all $\epsilon, \delta > 0$, for all $(\mathcal{S}, \mathcal{T})$, if S is a source generated sample set of size at least*

$$m \geqslant \frac{4}{\epsilon \delta C e} \left(\phi^{-1} \left(\frac{\epsilon}{4} \right) \sqrt{d} \right)^d$$

then, with probability at least $1 - \delta$ (over the choice of S), the target error of the Nearest Neighbor classifier is at most ϵ.

Proof. From the proof in Theorem 1, the error was bounded under two circumstances. As for (a), we apply Markov's inequality and get

$$\mathop{\mathbb{E}}_{S \sim \mathcal{S}^m} \left[2 \sum_{i : S \cap C_i = \varnothing} \mathcal{T}(C_i) \geqslant \frac{\epsilon}{2} \right] \leqslant \frac{4}{\epsilon \eta^d C m e} \tag{15}$$

Then for (b), we just set $2\phi(\frac{1}{\eta\sqrt{d}}) = \frac{\epsilon}{2}$, so $\eta = \frac{\sqrt{d}}{\phi^{-1}(\epsilon/4)}$. Finally, setting the probability to be smaller than δ yields that if

$$m \geqslant \frac{4}{\epsilon \delta C e} \left(\phi^{-1} \left(\frac{\epsilon}{4} \right) \sqrt{d} \right)^d$$

then with probability at least $1 - \delta$, the target error of the Nearest Neighbor classifier is at most ϵ. □

Theorem 5. *For two encoders $\mathbf{x}^a, \mathbf{x}^b$, a distribution $\mathcal{D}$ and a labeling function l, then*

$$\mathbf{x}^a \text{ is more clusterable than } \mathbf{x}^b \Leftrightarrow \begin{cases} \mathcal{L}_{\text{intra}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{intra}}(\mathbf{x}^b, \mathcal{D}, l) \\ \mathcal{L}_{\text{inter}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{inter}}(\mathbf{x}^b, \mathcal{D}, l) \end{cases}$$

Proof. $(\Rightarrow)$ There exists $\epsilon \in (0, 1)$, and $\lambda \in \{\lambda_1, \lambda_2\}$ with $\lambda_1 \lambda_2 < 0$, such that

$$\mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : |l(\mathbf{x}^a(\mathbf{z}_1), \mathbf{x}^a(\mathbf{z}_2)) - l(\mathbf{x}^a(\mathbf{z}_1), \mathbf{x}^a(\mathbf{z}_3))| - \epsilon \geqslant \lambda \|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\| \right)$$
$$\leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : |l(\mathbf{x}^b(\mathbf{z}_1), \mathbf{x}^b(\mathbf{z}_2)) - l(\mathbf{x}^b(\mathbf{z}_1), \mathbf{x}^b(\mathbf{z}_3))| - \epsilon \geqslant \lambda \|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\| \right)$$

When $l(\mathbf{z}_1, \mathbf{z}_2) = 1, l(\mathbf{z}_1, \mathbf{z}_3) = 1$, and $\lambda = \lambda_1 < 0$,

$$\mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\| \geqslant -\frac{\epsilon}{\lambda_1} \right) \leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\| \geqslant -\frac{\epsilon}{\lambda_1} \right).$$

So $\mathcal{L}_{\text{intra}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{intra}}(\mathbf{x}^b, \mathcal{D}, l)$. And let $l(\mathbf{z}_1, \mathbf{z}_2) = 1, l(\mathbf{z}_1, \mathbf{z}_3) = 0$, and $\lambda = \lambda_2 > 0$, then

$$\mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\| \leqslant \frac{1-\epsilon}{\lambda_1} \right) \leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\| \leqslant \frac{1-\epsilon}{\lambda_1} \right).$$

So $\mathcal{L}_{\text{inter}}(\mathbf{x}^a, \mathcal{D}, l) \leqslant \mathcal{L}_{\text{inter}}(\mathbf{x}^b, \mathcal{D}, l)$.

$(\Leftarrow)$ We have

$$\sum_{l(\mathbf{x}^a(\mathbf{z}_1), \mathbf{x}^a(\mathbf{z}_2)) = 1} \|\mathbf{x}^a(\mathbf{z}_1) - \mathbf{x}^a(\mathbf{z}_2)\| \leqslant \sum_{l(\mathbf{x}^b(\mathbf{z}_1), \mathbf{x}^b(\mathbf{z}_2)) = 1} \|\mathbf{x}^b(\mathbf{z}_1) - \mathbf{x}^b(\mathbf{z}_2)\|.$$

Denote C_1 as the mean value of $\mathcal{L}_{\text{intra}}(\mathbf{x}^b, \mathcal{D}, l)$, then

$$\mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\| \geqslant C_1 \right) \leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\| \geqslant C_1 \right).$$

In like manner, denote C_2 as the mean value of $\mathcal{L}_{\text{inter}}(\mathbf{x}^a, \mathcal{D}, l)$, then

$$\mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^a(\mathbf{z}_2) - \mathbf{x}^a(\mathbf{z}_3)\| \leqslant C_2 \right) \leqslant \mathop{\mathbb{P}}_{\mathbf{z}_1, \mathbf{z}_2 \sim \mathcal{D}} \left(\exists \mathbf{z}_3 : \|\mathbf{x}^b(\mathbf{z}_2) - \mathbf{x}^b(\mathbf{z}_3)\| \leqslant C_2 \right).$$

□

Theorem 6. *For two encoders $\mathbf{x}^a, \mathbf{x}^b$, a distribution $\mathcal{D}$, if η is a random variable and its support is a subset of $\mathbb{R}^+$, then*

$$\mathcal{L}_{\text{WR}}(\mathbf{x}^a, \mathcal{D}) \leqslant \mathcal{L}_{\text{WR}}(\mathbf{x}^b, \mathcal{D}) \Leftrightarrow \mathbb{E}\left[C_{\mathfrak{B}, \eta}(\mathcal{S}, \mathcal{D}; \mathbf{x}^a)\right] \geqslant \mathbb{E}\left[C_{\mathfrak{B}, \eta}(\mathcal{S}, \mathcal{D}; \mathbf{x}^b)\right]$$

Proof.

$$\begin{aligned}
& \mathcal{L}_{\mathrm{WR}}(\mathbf{x}^a,\mathcal{D}) \leqslant \mathcal{L}_{\mathrm{WR}}(\mathbf{x}^b,\mathcal{D}) \\
\Leftrightarrow\quad & \underset{\substack{\mathbf{z}_d\sim\mathcal{D}\\ \mathbf{z}_s\sim\mathcal{S}}}{\mathbb{P}}(\|\mathbf{x}^a(\mathbf{z}_d)-\mathbf{x}^a(\mathbf{z}_s)\|\leqslant\eta) \geqslant \underset{\substack{\mathbf{z}_d\sim\mathcal{D}\\ \mathbf{z}_s\sim\mathcal{S}}}{\mathbb{P}}(\|\mathbf{x}^b(\mathbf{z}_d)-\mathbf{x}^b(\mathbf{z}_s)\|\leqslant\eta) \\
\Leftrightarrow\quad & (\forall b_0\in\mathfrak{B})\ \underset{\substack{\mathbf{z}_d\sim\mathcal{D}\\ \mathbf{z}_s\sim\mathcal{S}}}{\mathbb{P}}(\mathbf{x}^a(\mathbf{z}_s)\in b_0|\mathbf{x}^a(\mathbf{z}_d)\in b_0) \geqslant \underset{\substack{\mathbf{z}_d\sim\mathcal{D}\\ \mathbf{z}_s\sim\mathcal{S}}}{\mathbb{P}}(\mathbf{x}^b(\mathbf{z}_s)\in b_0|\mathbf{x}^b(\mathbf{z}_d)\in b_0) \\
\Leftrightarrow\quad & \underset{\mathbf{z}_s\sim\mathcal{S}}{\mathbb{P}}(\mathbf{x}^a(\mathbf{z}_s)\in b_0|\mathcal{T}(b_0;\mathbf{x}^a)\geqslant\eta) \geqslant \underset{\mathbf{z}_s\sim\mathcal{S}}{\mathbb{P}}(\mathbf{x}^b(\mathbf{z}_s)\in b_0|\mathcal{T}(b_0;\mathbf{x}^b)\geqslant\eta) \\
\Leftrightarrow\quad & \mathbb{E}\left[C_{\mathfrak{B},\eta}(\mathcal{S},\mathcal{D};\mathbf{x}^a)\right] \geqslant \mathbb{E}\left[C_{\mathfrak{B},\eta}(\mathcal{S},\mathcal{D};\mathbf{x}^b)\right]
\end{aligned}$$

□

B Additional Experimental Details and Results

We present the structure of the paper in Figure 2 and the most important contributions in our work are Theorem 2 and 3, both of which aim to turn the abstract and somewhat too theoretical assumptions into practical loss functions. Although Theorem 1 seems like a straightforward extension of previous work [3], it plays a fundamental role in the paper. Through the DA-learnability shown in Theorem 1, we can see that the three assumptions imposed on the distribution of two domains in Section 3 are sufficient for solving the domain adaptive re-ID problem. In other words, the sufficiency of reinforcing the three assumptions in Section 4 is shown via Theorem 1.

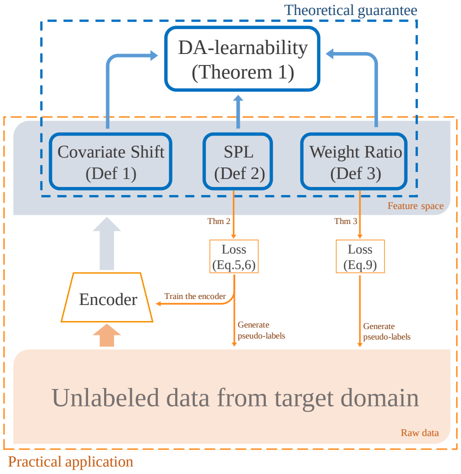
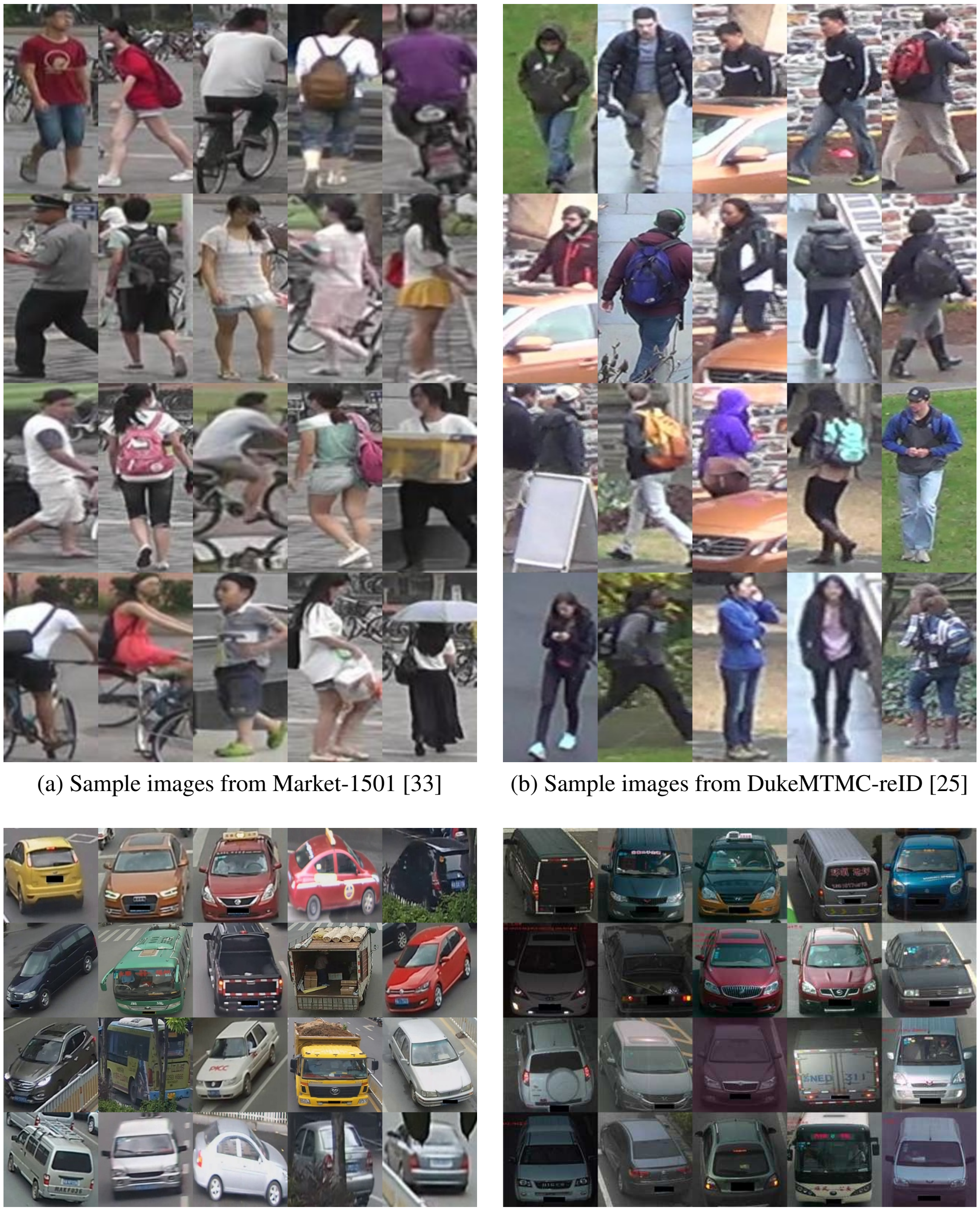


Figure 2: Structure of the paper.

B.1 Visualization of datasets and results

To understand the variations between different domains more clearly, Figure 3 presents some samples from the datasets used in our experiments. These datasets all have their own special characteristics. For instance, people riding a bicycle are common in Market-1501, while these people are rare in DukeMTMC-reID. More importantly, the images in these re-ID datasets are heavily related to the cameras, which means that the images contain information closely knitted together with the camera, such as background, viewpoints or lighting condition.

Moreover, we present some generated samples of SPGAN for vehicle re-ID. As shown in Figure 4, their image-image translation indeed works but fails to produce satisfactory re-ID results as person re-ID. This indicates that either their proposed generative method is not suitable for unsupervised domain adaptive vehicle re-ID, or their parameters need careful tuned for a new task.



(a) Sample images from Market-1501 [33]

(b) Sample images from DukeMTMC-reID [25]

(c) Sample images from VeRi-776 [16]

(d) Sample images from PKU-VehicleID [15]

Figure 3: Sample images from different datasets.

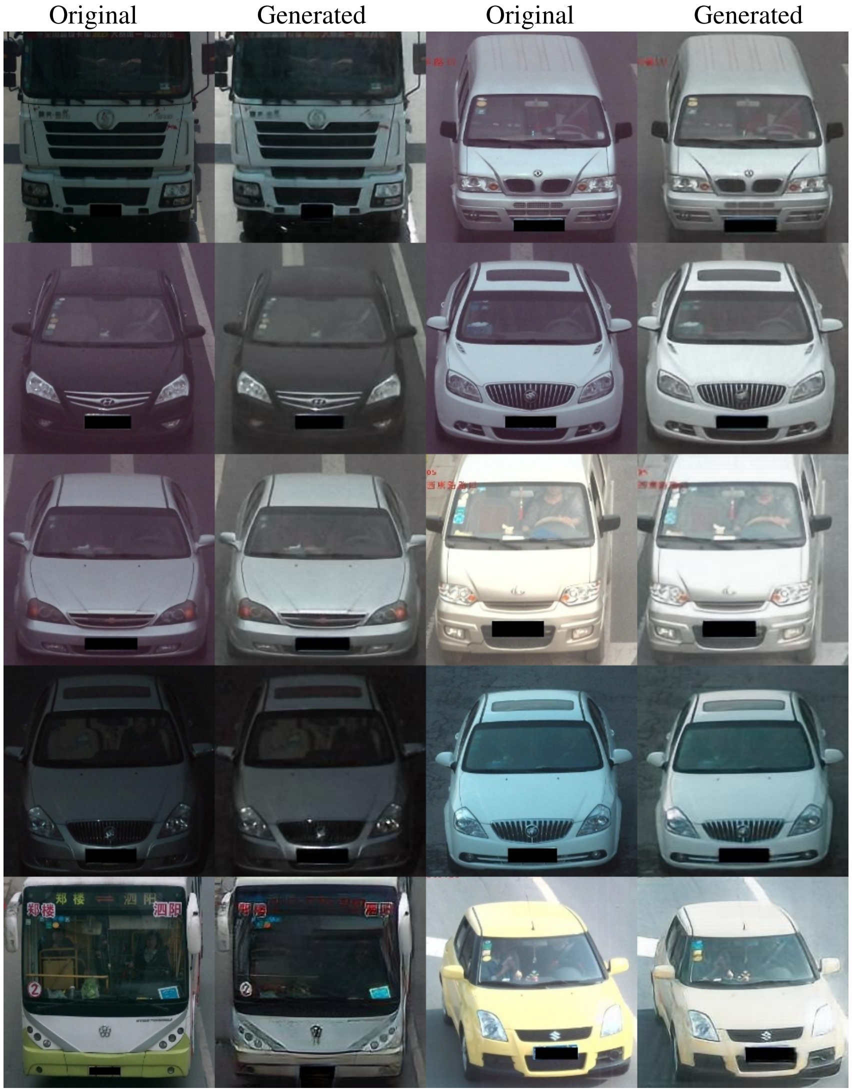


Figure 4: Sample images generated by SPGAN on vehicle datasets.

B.2 Encoder network

Basically, the encoder network is ResNet-50 [11] pre-trained on ImageNet and the whole network is presented in Figure 5.

Person re-ID. The size of input images is $256 \times 128 \times 3$, so the output of `conv5` is $8 \times 4 \times 2048$ and a average pool layer is added after `conv5` to have a output of size $1 \times 1 \times 2048$. We denote the output of this layer as `feat1`. During training on the source domain, `feat1` is connected to a fully-connected (`fc`) layer with output 2048, denoted `fc0`, then the 2048 `fc` layer is connected to a `fc` layer with output 751 (Market-1501) or 702 (DukeMTMC-reID). Let the output of finally `fc` layer be `fc1`. The loss functions are `Softmax(fc1)` and `Triplet(feat1)`, which are added directly (without extra balancing parameter). The model is trained by Adam optimizer [13]. Training parameters are set as follows: batch size 128 (PK sampling with P=16, K=8); maximum number epochs 70; learning rate 3e-4.

When training with data from target domain, there is no `fc1` layer and we use two triplet loss, that is `Triplet(feat1)` and `Triplet(fc0)`. The trick of using two triplet losses comes from [29]. The model is trained by stochastic gradient descent and in each iteration step we perform data augmentation (random flip and random erasing) on the data. Training parameters are set as

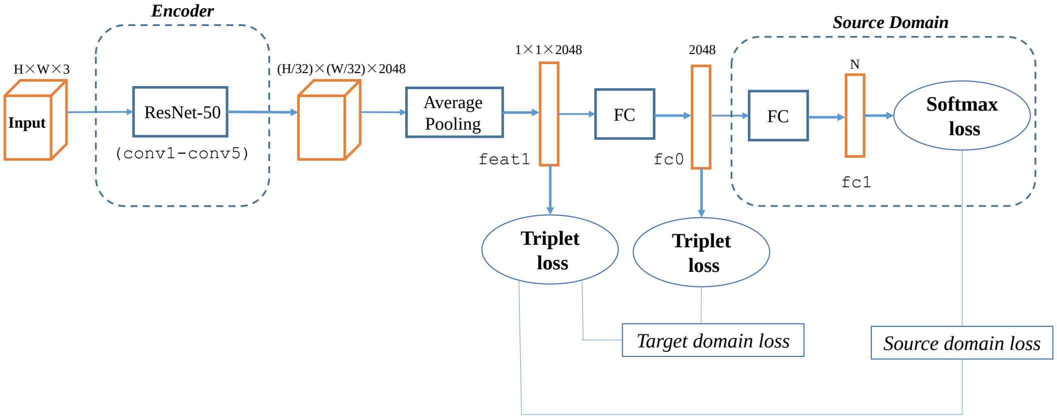


Figure 5: The architecture of our unsupervised domain adaptive network with ResNet-50 based encoder.

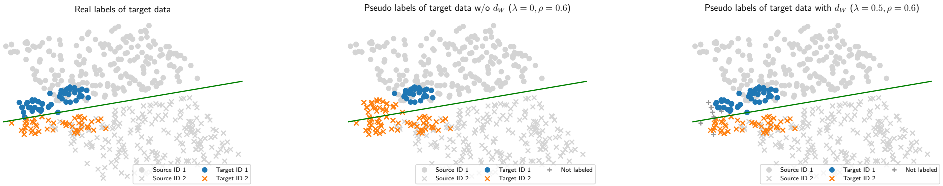


Figure 6: An example to show the effectiveness of d_{W}.

follows: batch size 128 (PK sampling with P=32, K=4); momentum 0.9; maximum number epochs 70; learning rate 6e-5. The networks are trained with two TITAN X GPUs.

Vehicle re-ID. All parameters including network architecture are same as person re-ID, except the size of input data. The input data here are resized to $224 \times 224 \times 3$ and the output of `conv5` is $7 \times 7 \times 2048$.

B.3 More results

Effectiveness of d_{W}. From Table 2, Table 3 and Figure 1, we observe that in a practical view, using d_{W} actually is not appealing. We think the reasons are two folds. Firstly, the effectiveness of d_{W} depends on the distribution of source and target domain. In Fig.(6), we design a simple example in 2D feature space to show the validity of d_{W}. In the left figure, the grey points denotes the extracted feature from source space and the colored points denotes the features of target data with real label. In the middle figure, we show the pseudo labels generated with `DBSCAN` when setting $\lambda = 0$, i.e., not using d_{W}. In the right figure, the results with d_{W} is shown. Comparing the middle figure and the right figure, we can see that d_{W} is important in such situation. The key idea in this demo is that those "easy" target points happen to be near the source data. Here, "easy" target points means the points belonging to the same ID are "close" in the extracted feature space with present encoder. This example can be also used for classification tasks since the Weight Ratio is a shared assumptions between our work and [3]. Secondly, d_{W} is derived from $\mathcal{L}_{\mathrm{WR}}$, but the potential value of $\mathcal{L}_{\mathrm{WR}}$ is not fully exploited in our algorithm. Thus, using d_{W} in practical application is not appealing. However, when not using d_{W}, the results are stable and good enough and already outperforms existing methods by large margin, which shows the power of the self-training scheme in domain adaptive re-ID problems. *For real applications, if the computation resources are limited, we recommend just setting $\lambda = 0$ and not making the effort to search for an optimal λ.*

Comparison of distance metrics. As for other contextual distance metrics, we test the performance of using the original Jaccard distance with or without d_{W} (also set $\lambda = 0.1$). For the Jaccard distance,

Table 4: Comparison of distance metrics.

Methods	DukeMTMC-reID→Market-1501			
	rank-1	rank-5	rank-10	mAP
Jaccard distance	63.8	80.0	85.3	37.1
Jaccard distance with d_{W}	**65.7**	**81.2**	**86.5**	**38.1**

Table 5: Comparison of clustering methods.

Distance Metrics	rank-1	rank-5	rank-10	mAP
Euclidean	**63.5**	**76.6**	**80.7**	**36.9**
Ours w/o d_{W}	62.4	74.6	78.9	35.2
Ours	62.4	74.5	78.8	35.6

we first compute the $k = 20$ nearest neighbor set and then compute the distance between the sets. Another conclusion is that taking d_{W} into consideration is also beneficial for Jaccard distance. However, both of the two distance metrics are worse than the self-training baseline, i.e., Euclidean distance. The reason is that the Jaccard distance only consider the nearest neighbor sets and therefore pairs without overlapping nearest neighbors will have a Jaccard distance 1, which is too strict to generate enough training pairs. The shortcoming also leads to a slow or even halted increasing of accuracy, for more details see the convergence comparison paragraph and Figure 1. As is shown in Table 4, k-reciprocal encoding employed in our method positively improve the performance of plain Jaccard distance.

Comparison of clustering methods. Due to the restrictions of a suitable clustering method, we only test a version with affinity propagation [9]. For task DukeMTMC-reID→Market-1501, we investigate the effectiveness of affinity propagation with other distance metrics. It is obvious that affinity propagation is not a proper clustering method for the reason that all data are used for clustering, which means it cannot avoid those pairs of low confidence. As shown in Table 5, a interesting fact is that with affinity propagation just using Euclidean distance is better than our proposed distance. The reason behind this phenomenon is that the number of IDs (clusters) generated by affinity propagation is much larger when using our proposed distance. In Figure 7, we show the number of IDs with respect to each iteration step. Using our distance leads to a larger number of clusters out of the reason that our distance will enlarge the gap between the dissimilar pairs, which is ought to be beneficial of getting rid of these helpless stray samples. However, affinity propagation is a clustering method that every sample is assigned to some cluster and therefore using our distance performs worse than Euclidean distance.

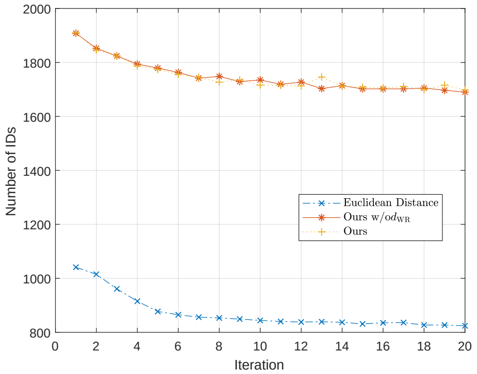


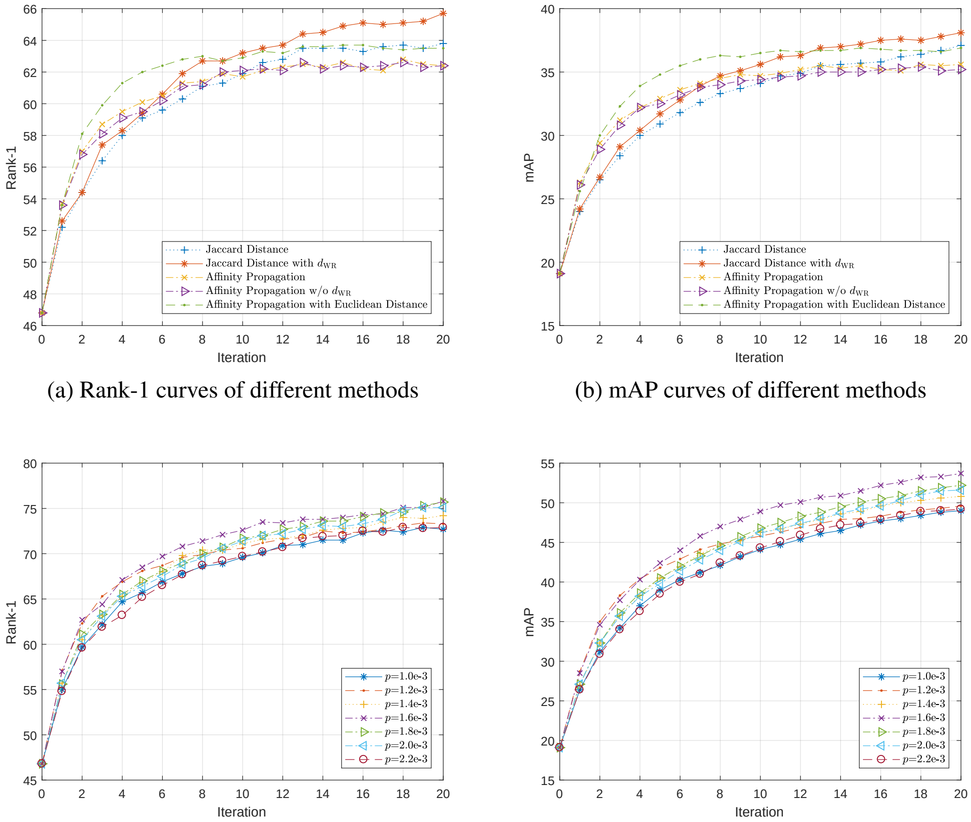
Figure 7: The number of IDs (clusters) on the target dataset of each iteration step when using affinity propagation.

Table 6: The impact of the parameter p on person re-ID (from DukeMTMC-reID to Market-1501).

	rank-1	rank-5	rank-10	mAP
$p = 1.0 \times 10^{-3}$	72.7	87.4	91.7	49.0
$p = 1.2 \times 10^{-3}$	73.3	86.0	89.5	49.6
$p = 1.4 \times 10^{-3}$	74.2	88.0	92.1	50.8
$p = 1.6 \times 10^{-3}$	**75.8**	**89.5**	**93.2**	**53.7**
$p = 1.8 \times 10^{-3}$	75.7	89.1	92.8	52.2
$p = 2.0 \times 10^{-3}$	75.1	88.7	92.3	51.6
$p = 2.2 \times 10^{-3}$	72.9	87.4	91.7	49.2

Parameters analysis Among all the parameters in our algorithm, the most influencing parameters are the percentage p and the balancing parameter λ. Since the influence of λ has been reported, here we perform experiments with a series of different p from DukeMTMC-reID to Market-1501 and the results are shown in Table 6. As we can see from the table, even a small change (2×10^{-4}) of p has a discernible impact on the final accuracy. It is because that we use large scale datasets and the number of all possible pairs from target datasets is large. Take Market-1501 as an example, the number of training images is 12,936, so the number of all data pairs is over 8×10^7. Thus a small change of p can cause a large change of the threshold.

Convergence comparison. In Figure 8, we use DukeMTMC-reID as source domain and Market-1501 as target domain and we first show the convergence results with different distance metric and clustering method in (a) and (b). Several conclusions can be drawn from the curves: First, we can see that the Jaccard distance based version becomes more stable after adding d_{W}; Second, the accuracy of the Jaccard distance based version almost stops increasing after 14 iterations, which is caused by the special property of Jaccard distance mentioned before; Third, using affinity propagation converges very fast and after about 8 iterations the accuracy stop increasing, which is caused by the inaccurate number of clusters and all the samples are used to train the network. Thus the loss functions fail to be minimized through sample selection step. Moreover, we show the results with different p in (c) and (d). It is obvious that all the curves have a similar convergence tendency, which demonstrates that our iteration process is robust with regard of the crucial parameter p.



(a) Rank-1 curves of different methods

(b) mAP curves of different methods

(c) Rank-1 curves with different p

(d) mAP curves with different p

Figure 8: Convergence comparison of different versions. We use DukeMTMC-reID as source domain and Market-1501 as target domain.