Pattern Recognition 122 (2022) 108293

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Who is closer: A computational method for domain gap evaluation

Xiaobin Liu*, Shiliang Zhang

Department of Computer Science, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 28 October 2020
Revised 2 August 2021
Accepted 31 August 2021
Available online 1 September 2021

Keywords:
Domain gap evaluation
CNN
Domain adaptive learning

ABSTRACT

Domain gaps between different datasets limit the generalization ability of CNN models. Precise evaluation on the domain gap has potential to assist the promotion of CNN generalization ability. This paper proposes a computational framework to evaluate gaps between different domains, *e.g.*, judging which one of source domains is closer to the target domain. Our model is based on the observation that, given a well-trained classifier on the source domain, the entropy of its classification scores of the output layer can be used as an indicator of the domain gap. For instance, smaller domain gap generally corresponds to smaller entropy of classification scores. To further boost the discriminative power in distinguishing domain gaps, a novel training strategy is proposed to supervise the model to produce smaller entropy on one source domain and larger entropy on other source domains. This supervision leads to an efficient and discriminative domain gap evaluation model. Extensive experiments on multiple datasets including faces, vehicles, fashions, and persons, *etc.* show that our method can reasonably measure domain gaps. We further conduct experiments on domain adaptive person ReID task and our method is adopted to pre-trained model selection, pre-trained model fusion, source dataset fusion, and source dataset selection. As shown in the experiments, our method substantially boosts the ReID accuracy. To the best of our knowledge, this is an original work focusing on computational domain gap evaluation. Our code is available at https://github.com/liu-xb/DomainGapEvaluation.

1. Introduction

Deep Neural Networks (DNNs) have exhibited impressive performance in many tasks, such as instance Re-Identification (ReID) [2,3,57–;59], semantic segmentation [4], object detection [5], object tracking [54–;56], *etc.* In those tasks, it is well known that, a well-trained DNN model may perform inferiorly on different target domains because of the domain gap issue. For example, the performance of ReID drops a lot on domains other than training domains as discussed in previous works [1,6,60]. It can also be inferred that, the scale of domain gaps varies among different datasets and larger domain gaps lead to more substantial performance degradations. For example, in person ReID task on *DukeMTMC-reID* [8] dataset, the model pre-trained on *Market-1501* [9] performs better than the model pre-trained on *CUHK03* [10]. Recently, many works try to tackle the domain gap issue from various aspects, *e.g.*, attention [7], adversarial leaning [11], Maximum Mean Discrepancy (MMD) based methods [12,13]. However, few works focus on the measurement of domain gap.

A computational method for domain gap evaluation is important, as it could assist many studies to boost the CNN generalization ability, such as transfer learning, domain adaptive learning, federated learning, *etc.* For example, domain adaptive person ReID task needs to choose an initial CNN model from several models pre-trained on different source datasets. In this case, domain gap evaluation can indicate which source domain is the closest one to the target domain. Choosing the model pre-trained on the closest source domain helps to promote the ReID performance on the target domain. Currently, the computational method for domain gap evaluation is still under-explored. This work is hence motivated to propose a computational method for domain gap evaluation.

It is not easy to directly train a domain gap evaluation model due to the lackage of domain gap annotation. We observe that, domain gap affects the distribution of classification outputs and large domain gap makes fuzzy classification scores. For example, for a well-trained classifier on the source domain, the testing data from a similar domain would produce a classification score vector with small entropy. As illustrated in Fig. 1, a classifier trained

* Corresponding author.
E-mail addresses: xbliu.vmc@pku.edu.cn (X. Liu), slzhang.jdl@pku.edu.cn (S. Zhang).

https://doi.org/10.1016/j.patcog.2021.108293
0031-3203/

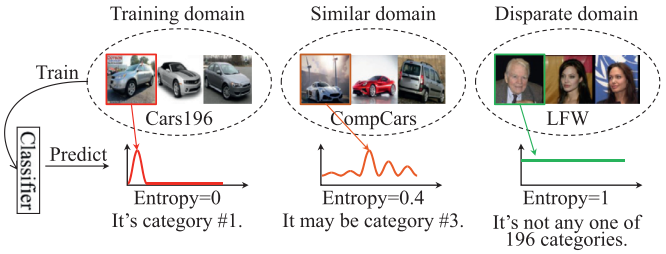

Fig. 1. Illustration of the classifier outputs on different domains. The classifier is trained on *Cars196* [14]. The curve below each domain shows the predicted probabilities over different categories in *Cars196*. It is clear that, the car dataset *CompCars* [15] produces smaller entropy than the face dataset *LFW* [16]. The entropy below each curve is normalized by the maximum entropy $\ln(C)$, where C denotes the number of categories in *Cars196*.

on *Cars196* [14] produces classification scores with small entropies when tested on *CompCars* [15]. When tested on the face dataset *LFW* [16], it produces classification scores with high entropies, indicating larger classification uncertainty. In other words, the output entropy of classification model can be used as an indicator of the domain gap between training and testing datasets.

Based on this observation, several methods for out-of-distribution detection [17–19] use the output probability of a classification model to detect whether a sample is from the training domain. However, these methods do not evaluate the gaps among different domains. As models in these methods is trained only on a single domain, they are only aware of the distribution of training domains. The outputs of different models trained on different source domains are independent to each other. Thus, the outputs by different classification models are not comparable to each other for domain gap evaluation.

In this paper, we target to learn domain gap evaluation models on different source domains with comparable outputs and awareness of domain gaps. To this end, a multi-task learning strategy is used to encourage the model to output different entropy on different domains. Specifically, a classification model is trained on a source domain with classification loss. In the meantime, samples from other source domains are also fed into the model and trained with proposed entropy loss, which encourages the output entropy to be the maximum value. After training, the entropy of the target domain can indicate the relationship of gaps between different training domains and the target domain. By comparing the output entropy of the target domain by different models with classification loss applied on different source domains, the relationship of gaps between different source domains and target domain can be obtained.

The proposed method is evaluated on multiple public datasets including *ImageNet* [20], *PACS* [21], *Office* [22], *DeepFashoin* [23], face datasets like *PubFig83* [24], vehicle datasets like *VeRi* [25], as well as person datasets like *MSMT17* [6]. Extensive experiments show that our method produces reasonable results on domain gap evaluation. For example, it indicates that semantically related datasets generally show smaller domain gaps than ones with distinct semantics. We further conduct experiments on domain adaptive person ReID task. As shown in experiments, our method helps to boost the performance on target domains, presenting the guidance for domain adaptive learning.

CNN suffers from substantial performance degradation due to the domain gap issue. It is still a critical task to boost the CNN generalization ability on different target domains. To the best of our knowledge, this is an original work studying the domain gap evaluation. It has potential to reveal the reason for domain gap, as well as to assist future studies on CNN structures and optimization methods towards better generalization ability.

2. Related work

2.1. Domain adaptation

Domain adaptation has been widely studied in many tasks from various aspects. Generative Adversarial Network (GAN) models are widely used to generate data that has approximate distribution with target domains. For example, Wei *et al.* [6] propose a GAN model to generate images that share the same distribution with target domains. Those generated images are used to bridge the domain gaps between different person datasets. Besides generating image, some researchers try to extract domain invariant features on different domains. For example, Cohen *et al.* [26] and Liu *et al.* [27] use adversarial learning to extract domain invariant features. There are also some methods based on MMD metric [12,28,29]. Despite the improvement on domain adaptation, how to measure the domain gap is still not yet studied.

Constraint on entropy is also commonly used on unlabeled dataset for domain adaptation task to obtain precise classification results on unlabeled data, *e.g.*, a loss also called "entropy loss" is used by Vu *et al.* [30] to minimize entropy on unlabeled data. Although our proposed method is also a constraint on entropy, it has different motivation, computation and results with previous methods. The motivation of reducing entropy on unlabeled data in previous methods is reducing the uncertainty and improving the accuracy of classification on unlabeled data. While, proposed entropy loss is motivated to improve the entropy on other source domains and make the model aware of domain gap between target domain and different source domains. Thus, the computations of proposed entropy loss are completely opposite with previous ones, *i.e.*, we *enlarge* the entropy on other source domains while previous methods *reduce* the entropy on unlabeled data. The training results are also different by our method and previous ones. The models trained by proposed entropy loss output maximum entropy on other source domains and are used for domain gap evaluation. While previous models perform classification on target domain.

Achille *et al.* [31] propose a TASK2VEC method to embed task into a vector to evaluate the distance between tasks, which is similar with our method. However, TASK2VEC focuses on task which is relative to the labels. Thus, different tasks on the same domain will have different embedding by TASK2VEC. While we focus on domains, *i.e.*, distribution of data. Thus, we only use the classification task in training to eliminate the difference caused by different tasks. Compared with TASK2VEC that needs labels in target domains, our method avoids the dependence on labels in target domains, making it more practical in unsupervised adaptive learning for person ReID task where target domains are unlabeled. Moreover, ReID task is not a training task and no direct loss is computed for this task, thus, the gradient based TASK2VEC is also not suitable.

Cheplygina *et al.* [32] review a number of articles to study whether medical data or nonmedical data is helpful for CT images recognition. Although intuitively medical data will be helpful, small scale of medical data makes CNN model easy to overfit. Thus, those nonmedical data will be helpful to avoid overfitting. There are also some researchers focus on detecting and understanding dataset bias [33–35]. However, domain bias detected by these methods can not precisely reflect the domain gap in transfer learning based on CNN. Compared with them, proposed domain gap evaluation method is evaluated on several transfer learning tasks.

2.2. Out-of-distribution detection

Out-of-distribution detection aims to detect whether a test sample is from the training domain. There exist many methods

to handle this task [17–19]. Devries *et al.* [19] propose a confidence estimation method to predict whether a sample comes from the training domain. Liang *et al.* [18] propose to use temperature scaling and perturbations on the input to separate distributions of softmax scores between in- and out-of-distribution samples. The above methods focus on detecting if a sample belongs to the training domain. However, they are not able to measure and compare the scale of domain gaps, *e.g.*, they are not able to tell which source domains are closer to a target domain. Differently, proposed method aims to measure gaps between domains.

2.3. Set-to-set distance

Many works have been proposed to compute the distance between image sets [36–38]. For example, Lu *et al.* [38] define the set-to-set distance as the average distance of samples in one set to the corresponding closest samples in another set. Zhu *et al.* [36] use the minimize distance between two hulls as the distance between two sets. Those methods are based on image pair similarity computed by shared models and mainly aim to achieve better image set recognition accuracy. Compared with set-to-set distance computation, domain gap evaluation is more complicated and challenging. Images in a set commonly belong a single category and different image sets belong to a same domain. However, an image domain commonly consists of multiple categories and a large number of images. This makes it time-consuming to compute image pairs similarity across domains and also hard to obtain a shared model for feature extraction on different domains. Moreover, domain gap evaluation lacks annotation, while set-to-set distance computation always has labels for training. Those differences make set-to-set distance computation methods not suitable for domain gap measurement. As a method to evaluate distance between two distributions, MMD is also used to evaluate the gap between two image datasets [12,28,29]. However, it is hard to obtain a reliable model for feature extraction on different domains.

3. Proposed approach

3.1. Problem formulation

Domain gap is an important factor that affects the performance of model when applied on different domains. Previous works only focus on bridging domain gaps yet fail to evaluate domain gaps. However, domain gap evaluation can also assist the studies on cross domain application, *e.g.*, domain adaptive learning for person ReID task. In this paper, we focus on evaluating domain gap. We consider domain gap as the obstacle to transferring discriminative ability of a model from the training dataset to another dataset. For instance, given two well-trained models M_1 and M_2 on two labeled source datasets S_1 and S_2, respectively, the discriminative ability of M_1 on the target dataset T is better than M_2 if the domain gap between S_1 and T is smaller than the gap between S_2 and T, and *vice versa*. The labeled dataset S can be denoted as follows:

$$S = \{(x_j^S, y_j^S) | y_j^S \in \{1, 2, \ldots, C^S\}, j = 1 \ldots N^S\}, \tag{1}$$

where x_j^S denotes the j-th image in S, y_j^S denotes the category label of x_j^S, N^S and C^S denote the number of samples and categories in S, respectively. The unlabeled target dataset T can be denoted as: $T = \{x_j^T | j = 1 \ldots N^T\}$, where x_j^T denotes the j-th sample in T and N^T denotes the number of samples in T. The domain gap between S and T is denoted as $G(T, S)$. Given multiple labeled source datasets $\{S_1, S_2, \ldots\}$ and the unlabeled target dataset T, this paper aims to compare the gaps between different source datasets and T.

Intuitively, we want to learn a domain gap evaluation function $\mathrm{F}_T(S)$ that computes $G(T, S)$. However, annotation of domain gaps is not available, hindering the learning of $\mathrm{F}_T(\cdot)$. To release the dependency on domain gap annotation, we propose to learn a domain gap comparison model for each source domain against other source domains, *i.e.*, learning $\{\mathrm{F}_{S_1/A_1}(T), \mathrm{F}_{S_2/A_2}(T), \ldots\}$ for $\{S_1, S_2, \ldots\}$, respectively, where $\{A_1, A_2, \ldots\}$ denote $\{\bigcup_{i\neq 1} S_i, \bigcup_{i\neq 2} S_i, \ldots\}$ and $\mathrm{F}_{S_i/A_i}(T)$ denotes the relative domain gap between S_i and T in the condition of comparing $G(T, S_i)$ and $G(T, A_i)$. Then, the comparison among $\{G(T, S_1), G(T, S_2), \ldots\}$ can be replaced with $\{\mathrm{F}_{S_1/A_1}(T), \mathrm{F}_{S_2/A_2}(T), \ldots\}$. To ensure the validity of this replacement, we propose a requirement that outputs of $\{\mathrm{F}_{S_1/A_1}(T), \mathrm{F}_{S_2/A_2}(T), \ldots\}$ should be comparable. Finally, the domain gap comparison between T and $\{S_1, S_2, \ldots\}$ can be achieved via the comparison between $\{\mathrm{F}_{S_1/A_1}(T), \mathrm{F}_{S_2/A_2}(T), \ldots\}$ with small function value indicating small domain gap.

It is worth noting that we are not aiming to compute absolute distance between different domains. Instead, we are trying to handle the task that comparing domain gaps between multiple optional source domains and the target domain, *i.e.*, we only considering the scenario of multiple source domains in this paper. This is reasonable because domain gap evaluation is often needed for comparing several optional source domains and choosing the best one for applications on the target domain in real-world application as shown in Section 4.9. Thus, proposed domain gap evaluation method computes the relative domain gap relationship instead of the absolute distance between domains to judge whether a source domain is closer to a target domain than other source domains or not. If only one source domain is provided, there is no need to compare domain gap for tasks shown in Section 4.9 as there is only one choice of source domain.

Without loss of generality, we take $\mathrm{F}_{S_1/A_1}(T)$ as an example to present the learning procedure. To ensure that the function $\mathrm{F}_{S_1/A_1}(T)$ can reflect the relationship of $G(T, S_1)$ and $G(T, A_1)$, S_1 and A_1 are both used in training with a multi-task learning manner. Specifically, a classification model with C^{S_1} outputs is used, which is denoted as φ_{S_1}. The classification loss ℓ_{cls} is applied on S_1, and the proposed entropy loss ℓ_{ent} is applied on A_1. The multi-task training strategy is illustrated in Fig. 2. The objective function is:

$$L = \ell_{cls} + \lambda \ell_{ent}, \tag{2}$$

where λ is the loss weight for ℓ_{ent}.

ℓ_{cls} encourages the model to right classify input images from S_1 which leads to entropy equalling 0. While ℓ_{ent} is proposed to encourage the model to predict equal probabilities over all C^{S_1} categories for input images from A_1, with output entropy being the maximum value, *i.e.*, $\ln(C^{S_1})$. Then, the output entropy of the model can be used as the indicator of whether T is close to S_1 or

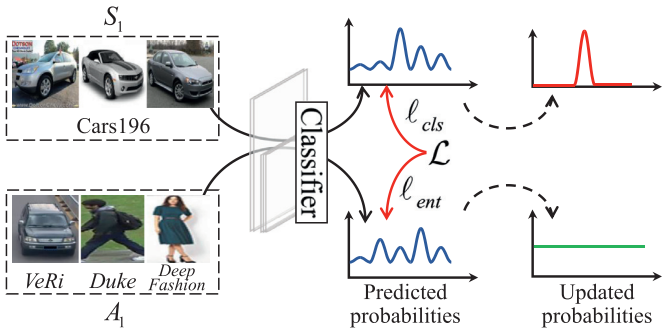

Fig. 2. Illustration of proposed multi-task learning strategy. Black arrows denote forward propagation, red arrows denote backward propagation, and black dotted arrows denote probabilities are updated after the model being trained by ℓ_{cls} and ℓ_{ent}. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

A_1, *i.e.*, small entropy indicates small $G(T, S_1)$ and large $G(T, A_1)$, and *vice versa*.

After training, the model φ_{S_1} is used to extract features of all images in T and entropy is computed on the outputs. As different source domains contain different number of categories, the maximum entropy of different classification model is also different. To make fair comparison between different models with different number of outputs, we propose to normalize the entropy by dividing the maximum entropy of each model, *e.g.*, the maximum output entropy of φ_{S_1} with C^{S_1} output is $\ln(C^{S_1})$ with probability for each category being $\frac{1}{C^{S_1}}$. For instance, given φ_{S_1} and φ_{S_2} has 100 and 10 outputs, respectively, the output probabilities of an image by two models are both 0.8 for the first category, 0.2 for the second category, and 0 for the rest categories. Then, the entropies by two models are both 0.722 and the normalized entropies are 0.109 and 0.217. As φ_{S_1} has more outputs than φ_{S_2}, outputs by φ_{S_1} provides more information gain than φ_{S_2} compared with random guess, and the input image is likely close to S_1. It is clear that the normalized entropy is more reasonable for comparison. Thus, $\mathrm{F}_{S_1/A_1}(T)$ is defined as the average normalized entropy on T and formulated as:

$$\mathrm{F}_{S_1/A_1}(T) = \frac{1}{N^T \ln(C^{S_1})} \sum_{j}^{N^T} \mathrm{Entropy}(p_j^T), \tag{3}$$

where $\mathrm{Entropy}(\cdot)$ denotes computing the entropy of output probability distribution and $p_j^T = \mathrm{Softmax}(\varphi_{S_1}(x_j^T))$ denotes the output probability of x_j^T. Functions $\{\mathrm{F}_{S_2/A_2}(\mathrm{T}), \mathrm{F}_{S3/A3}(T), \ldots\}$ are computed in a similar way. In the following, we will introduce the two components of objective function, respectively.

3.2. Classification loss

This section presents the computation of ℓ_{cls} in Eq. (2). Without loss of generality, we also take the computation of $\mathrm{F}_{S_1/A_1}(T)$ as an example, and ℓ_{cls} is applied on S_1. Cross entropy is adopted to formulate ℓ_{cls} as follows:

$$\ell_{cls} = -\frac{1}{N^{S_1}} \sum_{j=1}^{N^{S_1}} \ln(p_j^{S_1}(y_j^{S_1})), \tag{4}$$

where $p_j^{S_1}(y_j^{S_1})$ denotes the predicted probability for $y_j^{S_1}$-th category of the j-th sample in domain S_1, which is computed by softmax normalization as:

$$p_j^{S_1}(y_j^{S_1}) = \frac{\exp(o_j^{S_1}(y_j^{S_1}))}{\sum_{h=1}^{C^{S_1}} \exp(o_j^{S_1}(h))}, \tag{5}$$

where $o_j^{S_1}(h)$ denotes the h-th response value of the output of φ_{S_1} for $x_j^{S_1}$.

After being trained with ℓ_{cls}, model acquires categories knowledge of S_1 and is also aware of domain gaps between S_1 and other domains to some extent. As illustrated in Fig. 1, a classifier trained on *Cars196* will show small entropy on *CompCars* that is close to *Cars196*, and show large entropy on *LFW* that is far from *Cars196*. Therefore, the entropy of the output of a classification model can be used as a naive indicator of domain gaps as discussed in Section 1, *i.e.*, large entropies indicate large domain gaps and *vice versa*.

3.3. Entropy loss

Although model trained by classification loss can be aware of domain gaps, it may suffer from the weakness of the classifier. Moreover, as only one source domain is involved in training, the model is not aware of gaps between other source domains and the target domain.

To precisely evaluate gaps between different source domains and the target domain, we propose to additionally involve all other source domains to train the model by proposed entropy loss ℓ_{ent}, as shown in Fig. 2. Without loss of generality, we take $\mathrm{F}_{S_1/A_1}(T)$ as an example to present the learning procedure. To encourage the model to output differently on S_1 and A_1 in the aspect of predicted probability, ℓ_{ent} supervises the model to output probability with maximum entropy on A_1, *i.e.*, probability for each category is $\frac{1}{C^{S_1}}$ and the entropy is $\ln(C^{S_1})$. Therefore, we formulate ℓ_{ent} as follows:

$$\ell_{ent} = \ln(C^{S_1}) - \frac{1}{N^{A_1}} \sum_{j=1}^{N^{A_1}} \mathrm{Entropy}(p_j^{A_1}), \tag{6}$$

where $\ln(C^{S_1})$ is involved to keep the value of ℓ_{ent} positive, N^{A_1} denotes the number of images in A_1, $\mathrm{Entropy}(p_j^{A_1})$ denotes the entropy of the predicted probability $p_j^{A_1}$ for j-th sample in A_1, which is computed as:

$$\mathrm{Entropy}(p_j^{A_1}) = -\sum_{k=1}^{C^{S_1}} p_j^{A_1}(k) \ln(p_j^{A_1}(k)). \tag{7}$$

Models $\{\varphi_{S_2}, \varphi_{S3}, \ldots\}$ are also learned via multi-task learning by applying ℓ_{cls} on a source domain S_i and ℓ_{ent} on A_i, respectively. $\{\mathrm{F}_{S_2/A_2}(\cdot), \mathrm{F}_{S3/A3}(\cdot), \ldots\}$ are then computed following Eq. (3).

In training procedure, we feed samples from both S_1 and A_1 together to φ_{S_1} and apply ℓ_{cls} and ℓ_{ent} on samples from S_1 and A_1, respectively. ℓ_{cls} encourages output entropies to be 0 on S_1, and ℓ_{ent} encourages output entropies to be 1 on A_1 containing all the other source datasets. In this way, the training procedure defines on which domain the model should output zero entropy and on which domains the model should output maximum entropy. Thus, output entropies on other domains have the reference. Therefore, $\mathrm{F}_{S_1/A_1}(T)$ is aware of the relationship of $G(T, S_1)$ and $G(T, A_1)$: If S_1 is closer to T than all the domains in A_1, the output distribution of T will be close to the output distribution of S_1, and $\mathrm{F}_{S_1/A_1}(T)$ will tend to be a small value, *i.e.*, a smaller $\mathrm{F}_{S_1/A_1}(T)$ indicates a smaller $G(T, S_1)$ and a larger $G(T, A_1)$. While, if there exist one or more domains in A_1 that are closer to T than S_1, the output distribution of T will be close to the one of A_1, and $\mathrm{F}_{S_1/A_1}(T)$ will tend to be a large value, *i.e.*, a larger $\mathrm{F}_{S_1/A_1}(T)$ indicates a larger $G(T, S_1)$ and a smaller $G(T, A_1)$. Without ℓ_{ent}, model will be trained without the awareness of other source domains, and the output entropy will not reflect the domain gap relationship between them. Evaluation on ℓ_{ent} is shown in Section 4.7.1.

By comparing $\mathrm{F}_{S_1/A_1}(T)$ and $\mathrm{F}_{S_2/A_2}(T)$, $G(T, S_1)$ and $G(T, S_2)$ can be compared quantitatively. For instance, if $\mathrm{F}_{S_1/A_1}(T)$ is smaller than $\mathrm{F}_{S_2/A_2}(T)$, S_1 is closer to T than A_1, and S_2 is farther to T than $A2$. As A_1 contains $S2$ and A_2 contains S_1, it can be inferred that S_1 is closer to T than S_2. Thus, comparison between $\{G(T, S_1), G(T, S_2), \ldots\}$ can be achieved via the comparison between $\{\mathrm{F}_{S_1/A_1}(T), \mathrm{F}_{S_1/A_2}(T), \ldots\}$.

Discussion: Compared with previous domain distance computation methods that do not involve model training, *e.g.*, MMD, our method provides a reasonable training method without domain gap annotation. Our models are learned specially for domain gap evaluation, thus, they are aware of domain gap. While previous methods extract features by models that are not trained for domain gap evaluation, *e.g.*, models pre-trained on *ImageNet* for recognition. This makes previous methods not able to precisely evaluate domain gap.

In our method, entropy is used to evaluate the uncertainty of predicted probabilities. Other metrics for uncertainty estimation [17,19] can also be adopted. However, those metrics involve complicated computation for each sample, which is not efficient to

evaluate all the images in a dataset. Moreover, experiments show that proposed method is more reliable in domain gap evaluation.

Model trained by ℓ_{cls} and ℓ_{ent} is also able to detect out-of-distribution samples based on the entropy of them. If S_1 is very close to A_1 or even share the same domain with it, our method will become an out-of-distribution detection method. This character will be shown in experiments.

Previous researchers focus on bridging domain gap. Though domain gap can be inferred by performance of transfer learning, in real-world application, it is not available to evaluate the model on target unlabeled dataset, while proposed method does not need labels on target dataset.

3.4. Guidance on unsupervised learning

3.4.1. Pre-trained model and source dataset selection

In unsupervised domain adaptation scenario, sometimes only pre-trained models on different source domains are provided, while using source labeled data to train a new model is unavailable owing to time and memory consume. Then, we need to select a pre-trained model that has potential to perform the best on the target unlabeled dataset T with little cost. Intuitively, a model pre-trained on a domain closer to T will perform better on it. Given several labeled datasets, we compare and select the closest one by proposed domain gap evaluation method. Let $\{S_1, S_2, \ldots\}$ denote sorted domains, *i.e.*, S_1 is the closest domain to T, and S_2 is farther to T than S_1. Let $\{M_1, M_2, \ldots\}$ denote corresponding pre-trained models on $\{S_1, S_2, \ldots\}$. Then, M_1 is selected for the application on T. This method is evaluated in Section 4.9.1.

In domain adaptive learning scenario where source domains are also provided for training, we can also select the closest source domain for training based on proposed domain gap evaluation method, and this is evaluated in Section 4.9.3.

3.4.2. Pre-trained model and source dataset fusion

In addition to selecting a pre-trained model for application on T, we also propose to merge features of multiple pre-trained models with weights based on domain gaps. The weight for model M_1 is defined as w_i, which is computed as: $w_i = \frac{1}{(\mathrm{F}_{S_i/A_i}(T))^2}$. Then outputs of different models can be fused based on weights. Compared to fusing different models with equal weight, the proposed weighting strategy takes domain gaps into consideration. Model pre-trained on a closer domain has larger weight, and *vice versa*. This fusion strategy is more reasonable and experiments in Section 4.9.2 show that it boosts the performance on T. If multiple source labeled datasets are available for pre-training, a simple way to use these datasets together for model pre-training is merging them as a large dataset and then training the model on the merged dataset. However, this method ignores the gaps between T and different labeled datasets in model training. We propose that in model training, different datasets should have different weights in loss computation and model update. Specially, we propose to weight different training set based on $\{w_1, w_2, \ldots\}$. Experiments in Section 4.9.4 show that this method boosts the performance compared with equal weight.

4. Experiment

4.1. Dataset and implementation detail

Proposed model is first evaluated on several toy datasets including *PubFig83* [24], *LFW* [16], *DeepFashion* [23], *DukeMTMC-reID* [8], *Cars196* [14], *CompCars* [15], and *VeRi* [25]. To avoid the variance from different collection methods, we further evaluate the domain gaps between different semantic sub sets within *ImageNet* [20]. We then perform experiments on four person image datasets and two domain adaptation datasets, *i.e.*, *CUHK03* [10], *DukeMTMC-reID* [8], *MSMT17* [6], *Market-1501* [9], *Office* [22] and *PACS* [21], to evaluate proposed method and show the guidance on unsupervised domain adaptation. Involved datasets are briefly introduced as follows:

PubFig83 is a dataset of face images. It contains 13,838 images of 83 persons. *LFW* is also a dataset of face images with 5749 persons and 13,233 images. *DeepFashion* is a fashion dataset. In our experiment, we use the category and attribute prediction benchmark with 44 categories. *Cars196* is a dataset of cars images with 196 categories and 16,185 images from Internet. *CompCars* is also a dataset of vehicle images from Internet. In our experiments, 14,939 images are used. *VeRi* is a vehicle dataset collected from surveillance cameras and 37,778 images are used in our experiments. *ImageNet* is a large-scale classification dataset with 1000 categories and around 1000 images per category. *DukeMTMC-reID* is a dataset of person images collected from surveillance cameras in Duke University. It contains 36,411 images of 1812 persons from 8 cameras. In person ReID task, 16,522 images of 702 identities are selected for training and others are used for testing. 3368 images from testing set are selected as query images, and remaining 19,732 images are used as gallery images. For convenience, we use *Duke* to denote *DukeMTMC-reID* in the rest of this paper. *CUHK03* is also a person image dataset collected from the Chinese University of Hong Kong. It contains 7048 images of 1467 persons captured from 10 cameras. All images are used for training in this paper. *Market-1501* is also a person image dataset including 32,668 images of 1501 persons collected from 6 cameras at Tsinghua University. In person ReID task, 12,936 images of 751 identities are selected for training in this paper. For convenience, we use *Market* to denote *Market-1501* in the rest of this paper. *MSMT17* is a person image dataset including 126,411 images of 4101 persons collected from 15 cameras at Peking University. In person ReID task, 32,621 images of 1041 identities are selected for training in this paper. *Office* consists of 31 classes and 4110 images in 3 domains: *amazon, webcam*, and *dslr. PACS*covers 7 object categories and 4 domains, *i.e., Photo, Art Paintings, Cartoon* and *Sketches*. The number of images in each domain is 1,670, 2,048, 2,344, and 3,929, respectively.

We use ResNet50 [39] as the backbone structure and change the output number to the number of categories for corresponding source dataset. We randomly select 32 images for ℓ_{cls} and 32 images for ℓ_{ent} to form the training batch. Adam is used to train the model for totally 5 epochs. Learning rate is set to 0.00035. Loss weight λ is set to 0.1. Note that we only need few epochs for training as we do not chase a high classification accuracy. In experiments, it is observed that the performance and domain gap only change slightly when we repeat the training. This might be because that the training data is sufficient for model convergence and a stable performance. For example, on ReID task, we repeat the experiments in Table 5 for 10 times to compute the mean and standard deviation of each value. It is observed that the standard deviation is rather small. Thus, we do not report the standard deviation in other tables for neatness.

4.2. Evaluation on toy datasets

In this section, we use the proposed method to evaluate domain gaps between several toy datasets. Details are presented in the following.

Use *CompCars* as T: We compare the domain gaps between *CompCars* and four datasets including *Cars196, VeRi, DeepFashion*, and *Duke*. The comparison result is shown in Fig. 3. The values below axis are the outputs of function $\mathrm{F}_{S_i/A_i}(T)$ with *Cars196, VeRi, DeepFashion* and *Duke* as S_i, respectively. It can be observed that *Cars196* is the closest one to *CompCars*. Although *VeRi* is also a vehicle image dataset, *Cars196* is closer to *CompCars* than *VeRi* be-

Table 1
Comparison by using *CompCars* as T.

S_i	*Cars196*	*VeRi*	*DeepFashion*	*Duke*	*Market*	*MSMT*	*CUHK*	*LFW*	*PubFig83*
F_{S_i/A_i}	0.377	0.795	0.911	0.934	0.944	0.912	0.898	0.957	0.969

Table 2
Comparison by using *LFW* as T.

S_i	*PubFig83*	*DeepFashion*	*Duke*	*Cars196*	*Market*	*MSMT*	*CUHK*	*CompCars*	*VeRi*
F_{S_i/A_i}	0.454	0.755	0.935	0.954	0.970	0.892	0.980	0.985	0.991

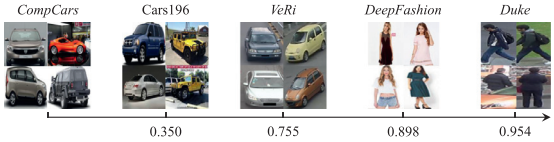

Fig. 3. Illustration of gaps between *CompCars* and different datasets. Gap enlarges from left to right. The values below axis are the outputs of function $F_{S_i/A_i}(T)$ with *Cars196, VeRi, DeepFashion* and *Duke* as S_i, respectively.

Fig. 4. Illustration of gaps between *LFW* to different datasets. Gap enlarges from left to right. The values below axis are the outputs of function $F_{S_i/A_i}(T)$ with *PubFig83, DeepFashion, Duke* and *Cars196* as S_i, respectively.

Table 3
Comparison by using *PubFig83* as S_1 and *LFW* as A_1. "Acc." denotes accuracy on test set of *PubFig83*.

A_1	Acc.	Entropy			
		PubFig83_test	*Cars196*	*DeepFashion*	*CUHK03*
None	0.91	0.155	0.859	0.867	0.887
LFW	0.90	0.142	1.00	1.00	0.999

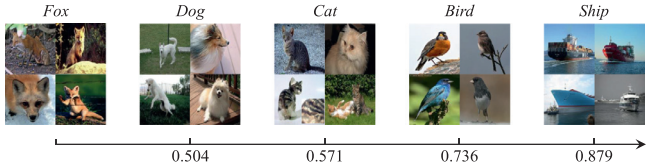

Fig. 5. Illustration of gaps between sub sets fox to others in *ImageNet*. Gap enlarges from left to right. The values below axis are the outputs of function $F_{S_i/A_i}(T)$ with *Dog, Cat, Bird* and *Ship* as S_i, respectively.

cause *VeRi* is collected from surveillance cameras while *Cars196* and *CompCars* are both collected from Internet. It can also be observed that, although *VeRi* is collected in a different way, it is still closer than other datasets with no vehicle images. This indicates that proposed method is robust to the variance in image-space. *DeepFashion* and *Duke* are both person image dataset, and *DeepFashion* is slightly closer to *CompCars* because it is also collected from Internet. The comparison among more datasets are shown in Table 1. Note that, we only choose several typical datasets to show the relationship of domain gaps in Fig. 3 because the challenging task is to judge whether *Cars196* or *VeRi* is closer to *CompCars*, and comparing domain gaps between *CompCars* and others is relatively easier. On the other hand, it can be observed in Table 11 that person ReID datasets such as Market, *CUHK03*, and *MSMT17* perform similarly with Duke. Face datasets *LFW* and *PubFig83* also perform similarly to each other. And these datasets are far from *CompCars*. Thus, only *Cars196, VeRi, DeepFashion*, and *Duke* are shown in Fig. 3.

Use *LFW* as T: The domain gaps between *LFW* and four datasets including *PubFig83, DeepFashion, Duke*, and *Cars196* are compared. The illustration of gaps is shown in Fig. 4. It can be observed that *PubFig83* is the closest one to *LFW* because it is also a person face dataset. It can also be observed that *DeepFashion* is closer than *Duke* because *DeepFashion* is also collected from Internet and faces in *DeepFashion* are clearer. *DeepFashion* and *Duke* are both closer than the car image dataset *Cars196* because they both contain human face in images. The comparison among more datasets are shown in Table 2. It can be observed that most datasets are far from *LFW*. Thus, we only choose several typical datasets to show the relationship of domain gaps in Fig. 4.

Use *PubFig83* as S_1 **and *LFW*as** A_1: As discussion in 3.3, when S_1 and A_1 are close to each other, our model will become an out-of-distribution detection method. To evaluate this character, we use *PubFig83* as S_1 and *LFW* as A_1 to train the model, then compute the normalized average entropy on the test set of *PubFig83, Cars196, DeepFashion*, and *CUHK03*. The results are summarized in Table 3. It can be observed that the model is aware of whether inputs are in the distribution of *PubFig83* as entropy of samples from the test set of *PubFig83* is much lower than the entropy of samples from other datasets. In the meantime, the model keeps the classification ability on the test set of *PubFig83*. This shows that our model is able to effectively detect out-of-distribution samples.

4.3. Evaluation on semantic sub sets in imagenet

Domain gaps between different datasets might be affected by different collection methods. To avoid the difference involved by collection method, in this section, we evaluate domain gaps between different semantic sub sets within a single dataset, *i.e., ImageNet* [20], to show that proposed method is also aware of semantic gap. We form 5 semantic sub sets with different numbers of categories, *i.e., Fox* with 4 categories, *Dog* with 118 categories, *Cat* with 13 categories, *Bird* with 59 categories, and *Ship* with 12 categories. The sub set *Fox* is set as T. Gaps between *Fox* and other sub sets are illustrated in Fig. 5.

It can be observed that *Dog* is the closest one to *Fox*. The next one is *Cat*, and *Ship* is the furthest one. We further compute the semantic similarity between *Fox* and *Dog, Cat*, and *Ship* by WordNet package in Python, respectively. The resulted similarities are 0.93, 0.86, and 0.44, which also indicates that *Dog* is the closest and *Ship* is furthest. This demonstrates that proposed method coincides with semantic similarity.

4.4. Evaluation on office

In this section, we apply the proposed method on *Office* dataset to evaluate gaps between three domains in *Office*. In this section, domain *webcam* is selected as T. We sample 400 images from each domain for evaluation. Based on proposed method, the outputs of domain gap evaluation functions are $F_{dslr/amazon}(webcam) =$

Table 4
Comparison on *webcam* with *amazon* and *dslr* as the source domain, respectively.

Source Domain		*amazon*	*dslr*
Domain Gap to *webcam*		0.998	0.821
Accuracy on *webcam*			
Method	Reference	*amazon*→*webcam*	*dslr*→*webcam*
DANN [40]	JMLR'16	0.730	0.964
ADDA [11]	CVPR'17	0.751	0.970
CDAN+E [41]	NeurIPS'18	0.941	0.986
PACET [42]	PR'19	0.908	0.976
ETD [43]	CVPR'20	0.921	1.000
DCAN [44]	PR'20	0.973	0.991
CAADA [45]	PR'20	0.802	0.971

Table 5
Comparison of performance of pre-trained models on *Duke*.M_1, M_2 and M_3 denote models pre-trained on *CUHK03, Market*, and *MSMT17*, respectively.

Source Domain	*CUHK*	*Market*	*MSMT17*
Domain Gap to *Duke*	0.931±0.003	0.629±0.002	0.522±0.003
Pre-trained Model	M_1	M_2	M_3
Rank1	0.323±0.004	0.360±0.005	**0.510±0.005**
mAP	0.127±0.002	0.171±0.002	**0.295±0.004**

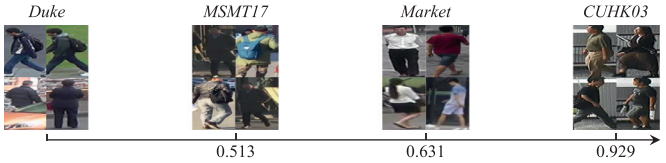

Fig. 6. Illustration of gaps between *Duke* to different datasets. Gap enlarges from left to right. The values below axis are the outputs of function $F_{S_i/A_i}(T)$ with *MSMT17, Market*, and *CUHK03* as S_i, respectively.

0.821 and $F_{amazon/dsl}(webcam) = 0.998$, respectively. This denotes that *dslr* is closer to *webcam* than *amazon*. It is reasonable as images in *amazon* are of high resolution and have white background, while *webcam* and *dsle* are collected by cameras with real-world background. Based on that, it can be inferred that the model pre-trained on *dslr* will perform better on *webcam* than the model pre-trained on *amazon*.

To evaluate this inference, we refer to several state-of-the-art methods in Table 4. It is clear that the model pre-trained on *dslr* outperforms the model pre-trained on *amazon* by a clear margin of each method. This coincides with our inference and also shows that proposed domain gap evaluation method is effective to select pre-trained model.

4.5. Evaluation on person image datasets

In this section, we first compare the gaps between *Duke* and other person image datasets including *Market, CUHK03*, and *MSMT17*. Then we use classification loss to train models on *Market, CUHK03*, and *MSMT17* to obtain pre-trained models, respectively. The transferred performance is used as the ground truth of domain gap evaluation. Following previous work [6], Rank1 and mAP accuracy are used to evaluate the person ReID performance.

The comparison of domain gaps between *Duke* and others is shown in Fig. 6. We can conclude that *MSMT17* is closer than *Market*, and they are both closer to *Duke* than *CUHK03*.

The comparison of person ReID performance on *Duke* is summarized in Table 5. M_1, M_2, and M_3 denote the models pre-trained on *CUHK03, Market, MSMT17*, respectively. It is obvious that the M_1 performs the worst and M_3 performs the best, which coincides with our inference. For example, M_1 achieves 0.312 in Rank1 accuracy, while M_2 and M_3 boost the Rank1 accuracy to 0.367 and 0.522, respectively. This demonstrates that our model can also precisely evaluate domain gap between person image datasets. It is worth noting that it is even hard for human to identify which person dataset is closer to *Duke* as images are both from surveillance cameras.

Table 6
Comparison on *Photo* with different source domains on *PACS*.

Source Domain		*Art*	*Cartoon*	*Sketches*
Domain Gap to *Photo*		0.445	0.578	0.887
Accuracy on *Photo*				
Method	Reference	*Art*→*Photo*	*Cartoon*→*Photo*	*Sketches*→*Photo*
JiGen [46]	ICCV'19	0.958	0.819	0.354
MMLD [47]	AAAI'20	0.967	0.888	0.432
RSC [48]	ECCV'20	0.946	0.842	0.484
CSD [49]	ICML'20	0.947	0.839	0.505

Table 7
Distances between *Duke* and other person image datasets by AD, EMD, and MMD, respectively. Numbers in bold font are the smallest gap in each row. Numbers in italics are the largest gap in each row.

		CUHK03	*Market*	*MSMT17*
f	AD	*0.269*	**0.224**	0.241
	EMD [50]	**0.616**	*0.624*	0.617
	MMD [12]	*0.378*	**0.263**	0.275
v	AD	**3.27**	3.54	*3.55*
	EMD [50]	1.53	*1.56*	**1.51**
	MMD [12]	0.126	*0.147*	**0.093**
Our Method		*0.929*	0.631	**0.513**

4.6. Evaluation on PACS

In this section, we apply the proposed method on *PACS* dataset to evaluate gaps between four domains in *PACS*. In this section, domain *Photo* is selected as T. We use 1500 images in each domain to train the model. The domain gap comparison between *Photo* and other domains is shown in Table 6. It is shown that *Art* is the closest one to *Photo* and *Sketches* is the farthest one.

To evaluate our inference on domain gap, we also refer to several state-of-the-art methods in Table 6. It is clear that the model pre-trained on *Art* outperforms models pre-trained on other domains by a clear margin. This coincides with our inference on domain gap comparison and also shows that domain gap evaluation could provide guidance for source domain selection.

4.7. Model analysis

4.7.1. Without ℓ_{ent}

To demonstrates that the proposed ℓ_{ent} makes the domain gap evaluation more precise, we perform experiments on different datasets without using ℓ_{ent} as follows.

On Toy Datasets: When using *CompCars* as T and only ℓ_{cls} in training, the average normalized entropies on *CompCars* by model trained on *Cars196, VeRi, DeepFashion*, and *Duke* are 0.475, 0.811, 0.455, and 0.912. This leads to the conclusion that *DeepFashion* is closer to *CompCars* than *Cars196* and *VeRi*, which goes against the fact that *Cars196* and *VeRi* are both vehicle datasets and should be closer to *CompCars*.

On Sub Sets in ImageNet: When using *Fox* as T and only ℓ_{cls} in training, the average normalized entropies on *Fox* by model trained on *Dog, Cat, Bird*, and *Ship* are 0.352, 0.281, 0.364, and 0.251. This leads to the conclusion that *Ship* is closer to *Fox* than *Dog* and *Cat*, which goes against the semantic similarities.

On person datasets: We compute the average normalized entropy with models pre-trained by only ℓ_{cls} on *Duke* to show the validity of proposed ℓ_{ent}. On *CUHK03, Market*, and *MSMT17*, the average normalized entropies are 0.806, 0.705, and 0.831, respectively. In this metric, *Market* is the closest one and *MSMT17* is the furthest one, which goes against the performance in Table 5.

These experiments demonstrate that the proposed ℓ_{ent} is important for precise domain gap evaluation.

4.7.2. Using classification model on T for domain gap evaluation

Assuming labels are available on T, a classification model can be trained on T, *i.e., Duke* in this section. Then, domain gaps can be evaluated by average normalized entropy on different source domains as in [17,19]. Here we compare proposed method with this simple method. We first train a classification model on *Duke*. Then average normalized entropies on *CUHK03, Market*, and *MSMT17* are 0.818, 0.760, 0.778, correspondingly. *Market* is the closest domain in this method, which goes against the performance comparison in Table 5. This further demonstrates the validity of proposed domain evaluation method.

4.7.3. Comparison with other metrics

We further compare proposed method with several widely used metrics including Average Distance (AD), Earth Mover's Distance (EMD), and Maximum Mean Discrepancy (MMD). We use a ResNet50 model pre-trained on *ImageNet* to extract pool5 features denoted as f, and also the final predicted probabilities vector denoted as v. We use Euclidean distance for a pair of pool5 features. For a pair of probabilities vectors, we use KL-divergence to compute the distance. Experimental results are summarized in Table 7. Numbers in bold font are the smallest gap in each row. It can be observed that these metrics are not able to effectively compare the domain gaps and the closest domain is falsely predicted in most cases. In the two cases where *MSMT17* shows the smallest distance, *CUHK03* shows smaller distance than *Market*, which goes against the performance comparison in Table 5. This indicates that these metrics cannot precisely evaluate domain gaps.

4.8. Factors affecting domain gap

This section studies which factors will affect domain gap. Specially, three factors that might affect domain gap are evaluated, *i.e.*, number of images, number of categories, and number of cameras in person dataset. Experiments show that proposed domain gap evaluation method is robust to those factors. Details are shown in the following.

4.8.1. Number of images and categories

This section studies whether the number of images and categories in source datasets will affect the domain gap.

In Section 4.3, different subsets contain different number of images. Specifically, as each category in *ImageNet* contains around 1000 images, sub sets *Dog, Cat, Bird* contain around 118,000 images, 13,000 images, and 59,000 images, respectively. *Dog* and *Bird* both have much more images than *Cat*. While *Dog* is closer to *Fox* and *Bird* is farther to *Fox* than *Cat*. This demonstrates that the number of images and categories in datasets does not affect the domain gap. This also demonstrates that proposed method is robust to the number images and categories in different source datasets.

In Section 4.4, 400 images from each source domain are randomly selected when comparing domain gaps. The comparison result coincides with domain adaptation performance by state-of-the-art methods that use all the images in source datasets. This also demonstrates that proposed method is robust to the number of used images.

In Section 4.5, *CUHK, Market*, and *MSMT17* have different number of images and categories. To eliminate the effect of different scales of datasets, we sample 5000 images from *CUHK, Market*, and *MSMT17* to re-compare the gaps between *Duke* in this section. The domain gap relationship is the same with the one in Section 4.5, *i.e., CUHK03* is the farthest and *MSMT17* is the closest. The Rank1 accuracies on *Duke* of models pre-trained on sub sets of *CUHK, Market*, and *MSMT17* are 0.162, 0.231, and 0.264, respectively. When we sample 10,000 images from each dataset for pre-training, the domain gap relationship is also the same one and the Rank1 accuracies of pre-trained models on *Duke* are 0.252, 0.302, and 0.436, respectively. This shows that results of our domain gap evaluation method coincide with the performance on the target dataset, and this further demonstrates that our method is robust to the number of images and categories in source datasets.

In Section 4.6, 1500 images from each source domain are randomly selected for domain gap evaluation. The evaluation result coincides with domain adaptation performance by several state-of-the-art methods that use all the images in source datasets as shown in Table 6. This further demonstrates that proposed method is robust to the number of used images.

4.8.2. Number of cameras in person datasets

Different person datasets are collected from different number of cameras. Specifically, *CUHK, Market*, and *MSMT17* are collected from 10 cameras, 8 cameras, and 15 cameras, respectively. Although *CUHK03* and *MSMT17* are both collected from more cameras than *Market, CUHK03* is farther to *Duke* and *MSMT17* is closer to *Duke* than *Market*. We further use images from random 8 cameras in *CUHK, Market*, and *MSMT17* to compare gaps between *Duke*. We find that the sub set of *MSMT17* is still the closest to *Duke* and *CUHK03* is still the farthest. Note that *Market* is collected from 8 cameras, so that sampling data from random 8 cameras results the whole dataset. The Rank1 accuracies on *Duke* of models pre-trained on sub sets of *CUHK03, Market*, and *MSMT17* are 0.264, 0.367, and 0.438, respectively. This shows that the results of proposed method coincide with the performance of pre-trained models.

4.9. Guidance on transfer learning

This section evaluates the guidance on transfer learning of our method on person image datasets. We evaluate the proposed pre-trained model selection strategy, pre-trained model fusion strategy, source dataset selection strategy, and dataset fusion strategy.

4.9.1. Pre-trained model selection

Based on the domain gap evaluation results in Section 4.5, it can be inferred that the model pre-trained on *MSMT17* performs the best on *Duke*, while the model pre-trained on *CUHK03* performs worst. The comparison of person ReID performance on *Duke* summarized in Table 5 shows that our model can provide effective guidance for unsupervised learning.

4.9.2. Pre-trained model fusion

This section merges features of different pre-trained models with proposed weighting method to show the guidance of domain gap evaluation. The weights are computed in Section 3.4. The performance comparison is shown in Table 8. "EWs" and "GBWs" denote equal weights and gap based weights, respectively. By merging $M_1 + M_2$, $M_1 + M_3$, $M_2 + M_3$, and $M_1 + M_2 + M_3$ with equal weight, the mAP accuracies on *Duke* are 0.197, 0.272, 0.295, and 0.281, respectively. By merging with proposed gap based weights, the mAP accuracies are 0.198, 0.299, 0.310, and 0.311, respectively. It is clear that merging models with proposed weighting method boosts the performance on the unlabeled dataset.

Table 8
Performance comparison of model fusion. "EWs" and "GBWs" denote equal weights and gap based weights, respectively. Each cell is in rank1/mAP format.

	M_1+M_2	M_1+M_3	M_2+M_3	$M_1+M_2+M_3$
EWs	0.386/0.197	0.496/0.272	0.510/0.295	0.499/0.281
GBWs	0.391/0.198	0.523/0.299	0.526/0.310	0.528/0.311
w	0.467+1	0.301+1	0.644+1	0.301+0.644+1

Table 9
Comparison of performance by different state-of-the-art domain adaptive methods on *MSMT17* with *Market* and *Duke* as source dataset, respectively.

Method	Reference	From *Market*		From *Duke*	
		Rank1	mAP	Rank1	mAP
PTGAN [6]	CVPR'18	0.102	0.029	0.118	0.033
ECN [3]	CVPR'19	0.253	0.085	0.302	0.102
SSG [1]	ICCV'19	0.316	0.132	0.322	0.133
SSG+ [1]	ICCV'19	0.376	0.166	0.416	0.183
GPP [51]	TPAMI'20	0.404	0.152	0.425	0.160
DIM+GLO [27]	ACMMM'20	0.497	0.207	0.565	0.244
MMT [52]	ICLR'20	0.525	0.263	0.588	0.297
D-MMD [53]	ECCV'20	0.291	0.135	0.344	0.153
Domain Gap to *MSMT17*	This paper	0.863		0.711	

4.9.3. Source dataset selection

We further use proposed method to guide the source dataset selection for domain adaptive person ReID task. In this task, a labeled dataset and an unlabeled dataset are provided for model training, and the target is improving performance on the unlabeled target dataset. As *Duke* and *Market* have a similar number of both images and identities, and many state-of-the-art methods have reported the performance on *MSMT17* by using *Duke* and *Market* as source dataset, respectively, in this section, *MSMT17* is used as the unlabeled dataset, and *Market* and *Duke* are used as labeled datasets, respectively. Moreover, as *Duke* and *Market* are both much smaller than *MSMT17*, this setting is closer to real-world application.

With proposed method, values of two functions $F_{Duke/Market}(MSMT17)$ and $F_{Market/Duke}(MSMT17)$ are 0.711 and 0.863, respectively. This indicates *Duke* is closer to *MSMT17* than *Market*. Then, it can be inferred that model trained with *Duke* as labeled dataset will perform better on *MSMT17*. To verify this inference, we show performances by different state-of-the-art domain adaptive methods in Table 9. "From *Market*" and "From *Duke*" denote *Market* and *Duke* are used as labeled training dataset, respectively. It can be observed that the model trained with *Duke* as the labeled dataset outperforms the model trained with *Market* as the labeled dataset by a clear margin in every method. Specially, although GPP outperforms SSG++ in same settings, SSG++ with *Duke* as labeled dataset even outperforms GPP with *Market* as labeled dataset. This indicates that source dataset selection matters for domain adaptive person ReID. This also shows that proposed domain gap evaluation method provides effective guidance for domain adaptive person ReID.

Table 10
Performance comparison of dataset fusion. "*CU.*", "*Ma.*", and "*MS.*" denote *CUHK03, Market*, and *MSMT17*, respectively. "EWs" and "GBWs" denote equal weights and gap based weights, respectively. Each cell is in rank1/mAP format.

	CU. + *Ma.*	*CU.*+*MS.*	*Ma.* + *MS.*	*CU.*+ *Ma.*+ *MS.*
EWs	0.425/0.211	0.523/0.293	0.536/0.308	0.555/0.336
GBWs	0.431/0.219	0.536/0.297	0.542/0.310	0.569/0.339
w	0.467+1	0.301+1	0.644+1	0.301+0.644+1

4.9.4. Source dataset fusion

This section evaluates proposed dataset fusion strategy for unsupervised learning. The mAP accuracies on *Duke* by merging different datasets are summarized in Table 10. It can be observed that fusing different datasets with gap based weights can improve the performance compared with using equal weight. This demonstrates the validity of proposed fusion strategy and the motivation of focusing on datasets closer to the target dataset.

4.9.5. End-to-end training by our method combined with different methods

In this section, we explore the end-to-end training by our domain gap evaluation method combined with domain adaptive training methods on person image datasets. *Market* and *Duke* are used as source domains and *MSMT17* is used as *T*. In model training, all source domains and the target domain are used and trained with different losses. We use released code to re-implement ECN [3] and GLO [27] on person datasets. The comparison is shown in Table 11.

It can be observed that, in the end-to-end training fashion, proposed domain evaluation method still works well and domain gap evaluation results still coincide with ReID performance, *i.e., Duke* is closer to *MSMT17* than *Market*. And the ReID performance is comparable with reported performance.

5. Discussion

5.1. Domain gap evaluation scenario

In this paper, domain gap evaluation is performed with multiple optional source domains provided. And our method computes relative gaps to compare different source domains, *i.e.*, our method tries to answer the question that which source domain is closer to the target one. This is reasonable as discussed in Section 3.1. This also coincides with the requirement for domain gap comparison of downstream tasks to choose the best source domain for applications on the target domain as discussed in Section 4.9.

If only one source domain is provided, there is no need to compare domain gap for tasks shown in Section 4.9 as there is only one choice of source domain. The absolute domain gap of a specific pair of domains might be also needed in other tasks, and we will further study how to compute the absolute domain gap in the future work.

Table 11
Comparison of different methods combined with ℓ_{ent}. Rank1 and mAP are computed on *MSMT17*. "ANE" denotes " Average Normalized Entropy".

	Market	*Duke*	*MSMT17*	Rank1	mAP	ANE on *MSMT17*
ECN [3]	ℓ_{cls}	ℓ_{ent}	$\mathcal{L}_{tgt}$ [3]	0.233	0.077	0.511
	ℓ_{ent}	ℓ_{cls}	$\mathcal{L}_{tgt}$ [3]	0.310	0.112	0.470
DIM+GLO [27]	ℓ_{cls}	ℓ_{ent}	$\mathcal{L}_{glo}$ [27]	0.479	0.187	0.404
	ℓ_{ent}	ℓ_{cls}	$\mathcal{L}_{glo}$ [27]	0.520	0.222	0.270

5.2. Advantages of our method

Proposed domain gap evaluation method is easy to implement and efficient in computation. Experiments show that our method coincides with common sense, semantic similarities, and transfer performance by many state-of-the-art methods on *Office, PACS, Duke* and *MSMT17*. We also show that domain gap evaluation has potential to assist transfer learning and unsupervised learning on ReID task.

Compared with previous methods that do not use labels or training, such as MMD, proposed method can evaluate domain gaps more precisely as shown in Table 7. This is because our method encodes semantic information provided by labels in source domains into the models in training. As MMD method do not perform training, it fails to encode semantic information as it is hard to obtain a reliable model for feature extraction on different domains. It is worth noting that the training of our method is performed offline and only cost around 1 h. As we do not chase a good classification performance on source domains, we do not need many training iterations. Thus, the training process does not affect the practical application of our method.

It is also worth noting that, training is always needed to encode semantic information provided in labels for semantic-level gap evaluation in current works such as TASK2VEC [31]. The advantage of our method compared with TASK2VEC [31] is that our method does not require any data in target domains, *i.e.*, training of our method is only performed on source domains. While TASK2VEC [31] needs data and label in target domains. Thus, our method is more practical than TASK2VEC [31].

5.3. Disadvantages and future work

Currently, our method needs to re-train models if a new source domain is added. Luckily, model re-training does not affect the referring time. However, we still hope to solve this issue in future study. A possible study direction is simplifying the training process, such as using one source domain as the reference and computing the relative gaps of different source domains compared to the reference. Another possible study direction could be computing the absolute domain gap in semantic level, which will be also studied in the future.

Conclusion

In this paper, we propose a computational method for domain gap evaluation. A multi-task training strategy with classification loss and proposed entropy loss is used to encourage the output entropy of the model to be different on different source domains. Then, the average normalized entropy of output in target dataset is used to evaluate domain gaps between the target domain and different source domains. Extensive experiments on multiple datasets demonstrate the validity of our method on evaluating domain gaps. Moreover, experiments also show that proposed domain gap evaluation method is instructive for applications on unlabeled datasets. This paper could raise the interest of researchers on domain gap evaluation, and this task also has potential to reveal the reason for domain gaps, as well as to assist future studies on CNN training and designing towards better generalization ability. In the future, we will further explore how to assistant transfer learning with domain gap evaluation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by Peng Cheng Laboratory, in part by The National Key Research and Development Program of China under Grant No. 2018YFE0118400, in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011, 61620106009, in part by Beijing Natural Science Foundation under Grant No. JQ18012.

References

[1] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, T.S. Huang, Self-similarity grouping: a simple unsupervised cross domain adaptation approach for person re-identification, ICCV, 2019.
[2] X. Liu, S. Zhang, X. Wang, R. Hong, Q. Tian, Group-group loss-based global-regional feature learning for vehicle re-identification, IEEE Trans. Image Process. 29 (2020) 2638–2652, doi:10.1109/TIP.2019.2950796.
[3] Z. Zhong, L. Zheng, Z. Luo, S. Li, Y. Yang, Invariance matters: Exemplar memory for domain adaptive person re-identification, CVPR, 2019.
[4] C. Liu, L.-C. Chen, F. Schroff, H. Adam, W. Hua, A.L. Yuille, L. Fei-Fei, Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation, CVPR, 2019.
[5] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, CVPR, 2019.
[6] L. Wei, S. Zhang, W. Gao, Q. Tian, Person transfer gan to bridge domain gap for person re-identification, CVPR, 2018.
[7] X. Wang, L. Li, W. Ye, M. Long, J. Wang, Transferable attention for domain adaptation, AAAI, 2019.
[8] Z. Zheng, L. Zheng, Y. Yang, Unlabeled samples generated by gan improve the person re-identification baseline *in vitro*, ICCV, 2017.
[9] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, ICCV, 2015.
[10] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: deep filter pairing neural network for person re-identification, CVPR, 2014.
[11] E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, CVPR, 2017.
[12] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, W. Zuo, Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation, CVPR, 2017.
[13] I.O. Tolstikhin, B.K. Sriperumbudur, B. Schölkopf, Minimax estimation of maximum mean discrepancy with radial kernels, NIPS, 2016.
[14] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, 3dRR-13, 2013.
[15] L. Yang, P. Luo, C. Change Loy, X. Tang, A large-scale car dataset for fine-grained categorization and verification, CVPR, 2015.
[16] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report, University of Massachusetts, Amherst, 2007.
[17] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, T.L. Willke, Out-of-distribution detection using an ensemble of self supervised leave-out classifiers, ECCV, 2018.
[18] S. Liang, Y. Li, R. Srikant, Enhancing the reliability of out-of-distribution image detection in neural networks, ICLR, 2018.
[19] T. DeVries, G.W. Taylor, Learning confidence for out-of-distribution detection in neural networks, arXiv preprint arXiv:1802.04865 (2018).
[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, CVPR, 2009.
[21] D. Li, Y. Yang, Y.-Z. Song, T.M. Hospedales, Deeper, broader and artier domain generalization, ICCV, 2017.
[22] K. Saenko, B. Kulis, M. Fritz, T. Darrell, Adapting visual category models to new domains, ECCV, 2010.
[23] Z. Liu, P. Luo, S. Qiu, X. Wang, X. Tang, Deepfashion: Powering robust clothes recognition and retrieval with rich annotations, CVPR, 2016.
[24] N. Pinto, Z. Stone, T. Zickler, D. Cox, Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook, CVPRW, 2011.
[25] X. Liu, W. Liu, H. Ma, H. Fu, Large-scale vehicle re-identification in urban surveillance videos, ICME, 2016.
[26] D. Cohen, B. Mitra, K. Hofmann, W.B. Croft, Cross domain regularization for neural ranking models using adversarial learning, ACM SIGIR, 2018.
[27] X. Liu, S. Zhang, Domain adaptive person re-identification via coupling optimization, ACM MM, 2020.
[28] Y. Chen, S. Song, S. Li, C. Wu, A graph embedding framework for maximum mean discrepancy-based domain adaptation algorithms, IEEE Trans. Image Process. 29 (2020) 199–213.
[29] H. Yan, Z. Li, Q. Wang, P. Li, Y. Xu, W. Zuo, Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation, IEEE Trans. Multimedia (2019). 1–1.
[30] T.-H. Vu, H. Jain, M. Bucher, M. Cord, P. Pérez, Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation, CVPR, 2019.
[31] A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C.C. Fowlkes, S. Soatto, P. Perona, Task2vec: Task embedding for meta-learning, ICCV, 2019.

[32] V. Cheplygina, Cats or cat scans: transfer learning from natural or medical image source data sets? Curr. Opin. Biomed. 9 (2019) 21–27.
[33] T. Tommasi, N. Patricia, B. Caputo, T. Tuytelaars, A deeper look at dataset bias, in: Domain adaptation in computer vision applications, Springer, 2017, pp. 37–55.
[34] A. Khademi, V. Honavar, Algorithmic bias in recidivism prediction: a causal perspective (student abstract), AAAI, 2020.
[35] B. Salimi, L. Rodriguez, B. Howe, D. Suciu, Interventional fairness: Causal database repair for algorithmic fairness, ICMD, 2019.
[36] P. Zhu, L. Zhang, W. Zuo, D. Zhang, From point to set: extend the learning of distance metrics, ICCV, 2013.
[37] B. Liu, L. Jing, J. Li, J. Yu, A. Gittens, M.W. Mahoney, Group collaborative representation for image set classification, Int. J. Comput. Vis. 127 (2) (2019) 181–206.
[38] J. Lu, G. Wang, W. Deng, P. Moulin, J. Zhou, Multi-manifold deep metric learning for image set classification, CVPR, 2015.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CVPR, 2016.
[40] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempitsky, Domain-adversarial training of neural networks, J. Mach. Learn. Res. 17 (1) (2016). 2096–2030.
[41] M. Long, Z. Cao, J. Wang, M.I. Jordan, Conditional adversarial domain adaptation, NeurIPS, 2018.
[42] J. Liang, R. He, Z. Sun, T. Tan, Exploring uncertainty in pseudo-label guided unsupervised domain adaptation, Pattern Recognit. 96 (2019) 106996.
[43] M. Li, Y.-M. Zhai, Y.-W. Luo, P.-F. Ge, C.-X. Ren, Enhanced transport distance for unsupervised domain adaptation, CVPR, 2020.
[44] Y. Chen, C. Yang, Y. Zhang, Y. Li, Deep conditional adaptation networks and label correlation transfer for unsupervised domain adaptation, Pattern Recognit. 98 (2020) 107072.
[45] M.M. Rahman, C. Fookes, M. Baktashmotlagh, S. Sridharan, Correlation-aware adversarial domain adaptation and generalization, Pattern Recognit. 100 (2020) 107124.
[46] F.M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, T. Tommasi, Domain generalization by solving jigsaw puzzles, CVPR, 2019.
[47] T. Matsuura, T. Harada, Domain generalization using a mixture of multiple latent domains, AAAI, 2020.
[48] Z. Huang, H. Wang, E.P. Xing, D. Huang, Self-challenging improves cross-domain generalization, ECCV, 2020.
[49] V. Piratla, P. Netrapalli, S. Sarawagi, Efficient domain generalization via common-specific low-rank decomposition, ICML, 2020.
[50] Y. Rubner, C. Tomasi, L.J. Guibas, The earth mover's distance as a metric for image retrieval, Int. J. Comput. Vis. 40 (2) (2000) 99–121.
[51] Z. Zhong, L. Zheng, Z. Luo, S. Li, Y. Yang, Learning to adapt invariance in memory for person re-identification, IEEE Trans. Pattern Anal. Mach. Intell. (2020).
[52] Y. Ge, D. Chen, H. Li, Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification, ICLR, 2020.
[53] D. Mekhazni, A. Bhuiyan, G. Ekladious, E. Granger, Unsupervised domain adaptation in the dissimilarity space for person re-identification, ECCV, 2020.
[54] C. Ma, C. Yang, F. Yang, Y. Zhuang, Z. Zhang, H. Jia, X. Xie, Trajectory factory: Tracklet cleaving and re-connection by deep siamese bi-gru for multiple object tracking, 2018 IEEE International Conference on Multimedia and Expo (ICME) (2018).
[55] C. Ma, et al., Deep association: End-to-end graph-based learning for multiple object tracking with conv-graph neural network, Proceedings of the 2019 on International Conference on Multimedia Retrieval (2019) 253–261.
[56] C. Ma, Deep Human-Interaction and Association by Graph-Based Learning for Multiple Object Tracking in the Wild, Int. J. Comput. Vis. 129 (2021) 1993–2010.
[57] X. Liu, S. Zhang, Graph consistency based mean-teaching for unsupervised domain adaptive person re-identification, IJCAI (2021).
[58] L. Wei, X. Liu, J. Li, S. Zhang, VP-ReID: Vehicle and person re-identification system, Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval (2018) 501–504.
[59] X. Liu, S. Zhang, Q. Huang, W. Gao, Ram: a region-aware deep model for vehicle re-identification, 2018 IEEE International Conference on Multimedia and Expo (ICME) (2018).
[60] J. Li, S. Zhang, Joint visual and temporal consistency for unsupervised domain adaptive person re-identification, European Conference on Computer Vision (2020).

Further Reading

R. Xu, G. Li, J. Yang, L. Lin, Larger norm more transferable: an adaptive feature norm approach for unsupervised domain adaptation, CVPR, 2019.
V.K. Kurmi, S. Kumar, V.P. Namboodiri, Attending to discriminative certainty for domain adaptation, CVPR, 2019.

Xiaobin Liu received the B.E. degree in intelligent science and technology from Nankai University, Tianjin, China, in 2016. He is now a Ph.D. candidate at the School of Electronic Engineering and Computer Science, Peking University, Beijing, China. His research area is computer vision and deep learning, with focus on image retrieval, vehicle and person re-identification and deep metric learning.

Shiliang Zhang received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2012. He was a Post-Doctoral Scientist with NEC Laboratories America and a Post-Doctoral Research Fellow with The University of Texas at San Antonio. He is currently a tenure-track Assistant Professor with the School of Electronic Engineering and Computer Science, Peking University. He has authored or co-authored over 70 papers in journals and conferences, including IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, ACM Multimedia, ICCV, and ECCV. His research interests include large-scale image retrieval and computer vision. He was a recipient of the Distinguished Young Scholar Fund of Beijing Natural Science Foundation, National 1000 Youth Talents Plan of China, Outstanding Doctoral Dissertation Awards from the Chinese Academy of Sciences and Chinese Computer Federation, the President Scholarship from the Chinese Academy of Sciences, the NEC Laboratories America Spot Recognition Award, and the Microsoft Research Fellowship. He was a recipient of the Top 10% Paper Award at the IEEE MMSP 2011. He serves as Associate Editor of IET Computer Vision, reviewer for 20+ Journals including ACM Computing Survey, IJCV, T-PAMI, T-IP, and TPC member for 10+ conferences including ICCV, CVPR, ECCV, ACM MM, IJCAI.