Learning to Adapt Invariance in Memory for Person Re-identification

Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, Yi Yang

Abstract—This work considers the problem of unsupervised domain adaptation in person re-identification (re-ID), which aims to transfer knowledge from the source domain to the target domain. Existing methods are primary to reduce the inter-domain shift between the domains, which however usually overlook the relations among target samples. This paper investigates into the intra-domain variations of the target domain and proposes a novel adaptation framework *w.r.t* three types of underlying invariance, *i.e.*, Exemplar-Invariance, Camera-Invariance, and Neighborhood-Invariance. Specifically, an exemplar memory is introduced to store features of samples, which can effectively and efficiently enforce the invariance constraints over the global dataset. We further present the Graph-based Positive Prediction (GPP) method to explore reliable neighbors for the target domain, which is built upon the memory and is trained on the source samples. Experiments demonstrate that 1) the three invariance properties are indispensable for effective domain adaptation, 2) the memory plays a key role in implementing invariance learning and improves the performance with limited extra computation cost, 3) GPP could facilitate the invariance learning and thus significantly improves the results, and 4) our approach produces new state-of-the-art adaptation accuracy on three re-ID large-scale benchmarks.

Index Terms—Person Re-identification, Domain Adaptation, Invariance Learning, Exemplar Memory, Graph-based Positive Prediction.

◆

1 INTRODUCTION

PERSON re-identification (re-ID) [59] is an image retrieval task, that aims at seeking matched persons of the query from a disjoint-camera database. The predominant methods have demonstrated dramatic performance when trained and tested on the same data distribution. However, they may suffer a significant degradation in the performance when evaluated on a different domain, due to dataset shifts from the changes of scenario, season, illumination, camera deployment, et al. It raises a domain adaptation problem that often encountered in real world applications and attracts increasing attention in the community [9], [10], [45], [64], [65]. In this work, we study the problem of unsupervised domain adaptation (UDA) in re-ID. The goal is to improve the generalization ability of models on a target domain, using a labeled source domain and an unlabeled target domain.

Conventional methods of UDA are mainly designed for a closed-set setting, where the source and target domains share a common label space, *i.e.* the classes of two domains are exactly the same. A popular approach is to align the feature distributions of both domains, but it does not readily

- *Z. Zhong is with the Department of Artificial Intelligence, Xiamen University, Xiamen 361005, China and also with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: zhunzhong007@gmail.com).*
- *L. Zheng is with the Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia (e-mail: liangzheng06@gmail.com).*
- *Z. Luo is with the Postdoc Center of Information and Communication Engineering, Xiamen University, Xiamen 361005, China (e-mail: zhiming.luo@xmu.edu.cn).*
- *S. Li (corresponding author) is with the Department of Artificial Intelligence, Xiamen University, Xiamen 361005, China (e-mail: szlig@xmu.edu.cn).*
- *Yi Yang is with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: yee.i.yang@gmail.com*

apply to the context of re-ID. Since domain adaptation in re-ID is a special open-set problem [4], [35], [38], where the source and target domains have completely disjoint classes/identities. For such label constraint, directly aligning the feature distributions of two domains will align the samples from different classes and may be detrimental to the adaptation accuracy.

To address the challenges of domain adaptive re-ID, recent works concentrate on aligning the source-target distributions in a common space, such as pixel-level space [9], [47] and attribute label space [27], [45]. Despite their success, these works only consider the overall inter-domain shift between the source and target domains, but largely overlook the intra-domain variations of the target domain. In the re-ID system, the intra-domain variations are important factors that affect the performance. Without considering the intra-domain variations of the target domain, an adapted model will produce poor performance, when the intra-domain variations in the target testing set are seriously different from the source domain.

In this work, we explicitly consider the intra-domain variations of the target domain and design our framework *w.r.t* three types of underlying invariance, *i.e.*, Exemplar-Invariance (EI), Camera-Invariance (CI), and Neighborhood-Invariance (NI), as described below.

Exemplar-Invariance (EI): The first property is motivated by the retrieval results of re-ID. Given a re-ID model trained on a labeled source training set, we evaluate it on a source/target testing set. On the one hand, we observe that the top-ranked retrieval results (both positive and negative samples) always are visually similar to the query when tested on the source set. A similar phenomenon is shown in image classification [51]. This indicates that the model has learned to distinguish persons by apparent similarity for the source domain. On the other hand, when tested on the target
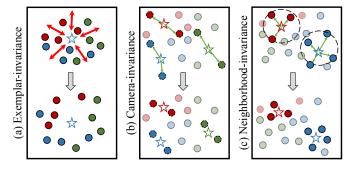
Fig. 1. Examples of three underlying properties of invariance. Colors indicate identities. (a) Exemplar-invariance: an input exemplar (denoted by ⋆) is enforced to be away from others. (b) Camera-invariance: an input exemplar (denoted by ⋆) and its CamStyle transferred images (with dashed outline) are encouraged to be close to each other. (c) Neighborhood-invariance: an input exemplar (denoted by ⋆) and its reliable neighbors (highlighted in dashed circle) are forced to be close to each other. Best viewed in color.

set, the top-ranked results often include many samples that are visually dissimilar to the query. This suggests that the ability of the model to distinguish persons by apparent similarity is degraded on the target domain. In reality, each person exemplar could differ significantly from others even shared the same identity. Therefore, it is possible to enable the model to capture the apparent representation by learning to distinguish individual exemplars. To achieve this goal, we introduce the exemplar-invariance (EI) to improve the discrimination ability of the model on the target domain, by encouraging each exemplar to be close to itself while away from others.

Camera-Invariance (CI): Camera style (CamStyle) difference is a critical factor for re-ID that can be clearly identified, since the appearance of a person may change largely under different cameras [66], [67]. Due to the camera deployments of the source and target domains are usually different, the model trained on the source domain may suffer from the variations caused by the target cameras. To address this problem, Zhong *et al.* [64] introduce camera-invariance (CI) by enforcing an target example and its corresponding Cam-Style transferred images to be close to each other. Inspired by them, we integrate the camera-invariance learning into our model by classifying an target example and its CamStyle counterparts to a same class.

Neighborhood-Invariance (NI): Apart from the easily identified camera variance, some other latent intra-domain variations are hard to explicitly discern without fine-grained labels, such as the changes of pose, view, and background. To overcome this difficulty, we attempt to generalize the model with the neighbors of target samples. Suppose we are given an appropriate model trained on the source and target domains, a target sample and its nearest-neighbors in the target set may share the same identity with a higher potential. Considering this trait, we introduce the neighborhood-invariance (NI) to learn a model that is more robust to overcome the latent intra-domain variations of the target domain. We accomplish this constraint by encouraging an exemplar and its reliable neighbors to be close to each other. Examples of the three types of invariance are illustrated in Fig. 1.

Intuitively, a straightforward way to implement the

three invariance properties is to constrain them with contrastive/triplet loss [14], [19] within a training mini-batch. However, the number of samples in a mini-batch is relatively small compared with the entire training set. In this manner, it is difficult to form a mini-batch with ideal examples, and the overall relations between training samples cannot be considered thoroughly during the network adaptation procedure. To deal with this issue, we develop a novel framework to effectively accommodate the three invariance properties for domain adaptive re-ID. Specifically, we introduce an exemplar memory module into the network to store the up-to-date representations of all training samples. The memory enables the network to enforce the invariance constraints over the entire/global target training data instead of the current mini-batch. With the memory, the invariance learning of the target domain can be effectively implemented with a non-parametric classification loss, considering each target sample as an individual class.

In our previous work [65], we directly select top-k nearest neighbors from the memory for the learning of NI. This straightforward strategy ignores the underlying relations between samples in the memory. As a result, the similarity estimation of hard samples may not be accurate when the model has inferior discriminative ability. As a notable extension of our previous work [65], we propose a graph-based positive prediction (GPP) approach to address this problem, thereby promoting the invariance learning. GPP is built upon the memory module and designed by graph convolutional networks (GCNs), which aims to predict positive neighbors from the memory for a training target sample. In addition to the target memory, we also construct a memory for saving features of the source domain. This enables us to imitate the neighbor exploring process of target invariance learning and thus learn GPP on the labeled source domain. The learned GPP is then directly applied to the unlabeled target domain for facilitating the learning of NI.

In summary, our contribution is as follow:

- This work comprehensively investigates the intra-domain variations of the target domain and studies three underlying properties of target invariance. The experiment demonstrates that the three properties are indispensable for improving the transferable ability of the model in the context of re-ID.
- This work proposes a novel framework equipped with a memory module that can effectively enforce the three constraints into the network. The memory enables us to fully exploit the sample relations over the whole training set instead of the mini-batch. With the memory, the performance can be significantly improved, requiring very limited extra computation cost and GPU memory.
- This work introduces a Graph-based Positive Prediction (GPP) approach to leverage the relationships between candidate neighbors and infer accurate positive neighbors for the training target sample. The experiment shows that GPP is beneficial to the learning of neighborhood-invariance and could consistently improve the results, especially the mAP.
- In addition, we analyze the mechanism of the three invariance properties, which helps us to understand

- how these three properties encourage the network to adapt to the target domain.
- Experiments demonstrate the effectiveness and superiority of the proposed method over the state-of-the-art UDA approaches. Our results outperform state of the art by a large margin on three datasets, including Market-1501, DukeMTMC-reID and MSMT17.

2 RELATED WORK

Unsupervised domain adaptation. This work is closely related to unsupervised domain adaptation (UDA). In classical UDA, methods are designed under the assumption of the closed-set scenario, where the classes of the source and target domains are precisely the same. For this problem, recent popular approaches are mainly focused on distribution alignment learning, which attempts to reduce the domain discrepancy by using Maximum Mean Discrepancy (MMD) minimization [13], [28], [53] or domain adversarial training [2], [43]. However, these methods are usually not applicable to the open-set scenario, where unknown classes exist in the source/target domain. Since samples of unknown classes should not be aligned with the ones of known classes. It raises to the problem of open set UDA, introduced by Busto and Grall [4]. To tackle this problem, Busto and Grall [4] develop a method to learn a mapping from the source domain to the target domain by jointly predicting unknown-class target samples and discarding them. Saito *et al.* [35] introduce an adversarial learning framework to separate target samples into known and unknown classes. Meanwhile, unknown classes are rejected during feature alignment. Recently, Sohn *et al.* [38] consider a more challenging setting of open-set UDA, where the source and target domain belong disjoint label spaces, *i.e.* the classes of two domains are completely different. In practice, many tasks match such setting, *e.g.*, cross-ethnicity face recognition and UDA in re-ID considered in our work. To address this problem, Sohn *et al.* first reformulate the disjoint classification task to a binary verification one. Then, a Feature Transfer Network (FTN) is proposed to transfer the source feature space to a new space and align with the target feature space. In this paper, we study this problem in the aspect of intra-domain variations in the target domain and propose an effective framework to improve the generalization ability of the model by target invariance learning.

Unsupervised person re-identification (re-ID). Recent supervised methods have made great achievements in re-ID [24], [41], [42], [60], [62], benefiting from rich-labeled data and the increasing ability of deep networks [17], [29]. However, the performance of these strong methods may have a large drop when tested on an unseen (target) dataset, due to the dataset shift. To address this problem, many hand-craft features based methods are designed for unsupervised re-ID, such as ELF [12], LOMO [25] and BOW [58]. These methods can be directly applied to any dataset without training, but fail to obtain fulfilling performance on large-scale, complex scenarios. Although labeling re-ID data is difficult, it is relatively easy to collect sufficient unlabeled data in the target domain. Recently, many works are proposed to transfer the knowledge of a labeled data to an unlabeled one. These works mainly can be divided

into two categories: 1) discovering pseudo labels for target dataset, and 2) reducing the source-target discrepancy in a common label space. For the first category, methods use a labeled source dataset to initialize a re-ID model and explore pseudo labels for target dataset based on clustering [10], [55], associating label with labeled source dataset [57], assigning label with nearest-neighbors [5], [22], [23], [54], or regarding camera style counterparts as positive sample [64]. These methods are closely related to our work in that using the relationship between target samples to refine the model. The main difference is that our work comprehensively considers three latent invariance constraints and enforces them over the entire dataset. We show the mutual benefit among the three invariance properties and the effectiveness of exploiting the global sample relationship. Methods of the second category attempt to align the distributions of the source and target domains in a common space, such as image-level space [1], [9], [47], [67] and attribute label space [27], [45]. These methods only consider the overall discrepancy between the source and target domains, but largely ignore the intra-domain variations in the target domain. In this work, we explicitly consider the invariance properties in the target domain to address the problem of domain adaptive re-ID.

Neural networks with augmented-memory. Learning neural networks with augmented-memory is proposed to address various tasks, such as question answering [40], [48], few-shot learning [36], [44], and video understanding [49]. The augmented-memory enables the networks to store the intermediate knowledge into a structural and addressable table. The community of learning with augmented-memory mainly can be divided into two categories. The first category aims to augment neural networks with a fully differentiable memory module, such as Neural Turing Machine [11] and Memory Networks [48]. Memory Networks introduce a long-term memory component that can be read and written. The memory is utilized to query and retrieve fact for the task of question answering. Another category focus on developing a non-parametric memory [44], [50], [51], [52] which can directly save the features of samples into a feature bank and be updated during training. These approaches then use the attention mechanism to calculate the similarities between the query and instances in the memory. We draw inspiration from these methods and develop a memory-based framework for unsupervised domain adaptive re-ID.

Graph convolutional network (GCN). In real-world applications, data are usually represented in the form of graphs or networks, such as knowledge graphs and social networks. Recently, many works concentrate on extending the deep convolutional neural network for handling graph data. In light of the objectives on graph data, these works are generally divided into spectral based methods [3], [7], [18], [21] and spatial based methods [15], [30], [32], [37], [46]. The spectral based methods are developed based on graph spectral theory. These methods usually handle the entire graph to learn a domain-dependent graph network, leading them difficult to apply on large graphs and can not be applied to a different structure. The spatial based methods extend the convolution operation of CNNs to the graph structure. The graph convolution is directly performed based on the graph nodes and their spatial neighbors. This work is mostly
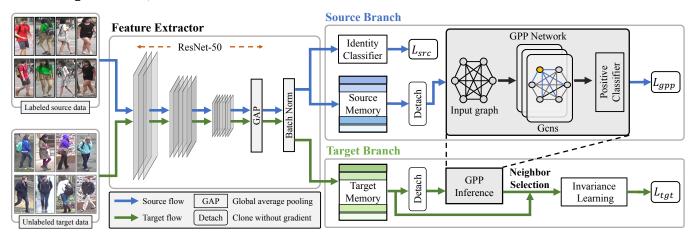
Fig. 2. The framework of the proposed method. During training, the inputs are drawn from the labeled source domain and the unlabeled domain. We feed-forward the inputs into the feature extractor to obtain the up-to-date representations. Subsequently, two branches are designed to optimize the framework with the source data and the target data, respectively. In each branch, we introduce a memory to store the features of corresponding domain data. The source branch aims to learn basic representation for the model with identity classification loss L_{scr}, as well as, to learn a graph-based positive prediction (GPP) network with binary classification loss L_{gpp}. For training of GPP, we first select top-ranked features of the input from the source memory. Then, the selected features are regarded as candidate neighbors and are used to train the GPP network. The GPP network is trained to predict the positive and negative neighbors of the input. The target branch attempts to enforce the invariance learning on the target data. The invariance learning loss L_{tgt} is calculated by estimating the similarities between the target sample and whole features in the target memory. In addition, we employ the GPP network to infer reliable positive neighbors for the target sample, thereby facilitating the invariance learning. In testing, the L2-normalized output of the global average pooling (GAP) is used as the feature of an image.

inspired by spatial based methods and proposes a graph-based positive prediction (GPP) approach to predict reliable neighbors across graphs formed by different candidates. The proposed GPP is also related to LBFC [46] and SGGNN [37], which are designed for face clustering and re-ID re-ranking, respectively. The main difference is that GPP is designed to explore reliable neighbors that contribute to generalize re-ID networks. Importantly, GPP is trained on hard negative samples selected from the whole dataset, while LBFC and SGGNN are trained with limited samples from a mini-batch.

3 THE PROPOSED METHOD

This paper aims to address the problem of unsupervised domain adaptation (UDA) in person re-identification (re-ID). In the context of UDA in re-ID, we are provided with a fully labeled source domain $\{X_s, Y_s\}$ and an unlabeled target domain X_t. The source domain includes N_s person images of M identities. Each person image x_i^s is associated with an identity y_i^s. The target domain contains N_t person images. The identity annotation of the target domain is not available. In general, the source and target domains are drawn from different distributions, and the identities in the source and target domains are completely different. In this paper, our goal is to learn a transferable deep re-ID model that generalizes well on the target testing set. The labeled source domain and unlabeled target domain are exploited during training.

3.1 Overview of Framework

The framework of the proposed method is illustrated in Fig. 2. In our method, the inputs are sampled from the labeled source domain and the unlabeled target domain. The inputs are first fed-forward into the feature extractor to obtain the up-to-date features. The feature extractor is composed of convolutional layers, global average pooling

(GAP) [26] and a batch normalization layer [20]. The convolutional layers are the residual blocks of ResNet-50 [17] pretrained on ImageNet [8]. The output of feature extractor is 2,048 dimensional. Subsequently, the source branch and the target branch are developed for training our model with the source data and the target data, respectively. Each branch includes an exemplar memory module. The memory module is served as a feature-storage that saves the up-to-date output of feature extractor for each source/target sample. The aims of the source branch are twofold. On the one hand, we use the identity classifier with cross-entropy loss to learn basic representations for the feature extractor. On the other hand, we first select the top-ranked features of the input from the source memory. These top-ranked features are regarded as candidate neighbors and are utilized to learn a Graph-based Positive Prediction (GPP) network. The GPP network consists several graph convolution layers [21] and a positive classifier, which aims to predict the probability of a candidate that belongs to a positive sample of the input. The target branch is designed to enforce the proposed three properties of invariance on unlabeled target data, *i.e.* exemplar-, camera- and neighborhood- invariance. Given a target sample, the invariance learning loss is calculated by estimating the similarities between the target sample and whole target samples in the target memory. During invariance learning, we use the learned GPP to infer reliable positive neighbors of the target sample. These neighbors are selected from the candidates of the target memory, according to the probabilities obtained by the positive classifier of GPP. Note that, the GPP network is trained with only the source data. Besides, the loss of the GPP network is not utilized to update the feature extractor.

3.2 Supervised Learning for Source Domain

Given the labeled source data, we are able to learn a basic discriminative model in a supervised way. In this paper,

we treat the training process of the source domain as an identity classification problem [59]. Thus, the supervised loss of the source domain is calculated by cross-entropy loss, formulated as,

$$\mathcal{L}_{src} = -\log p(y_i^s | x_i^s), \qquad (1)$$

where $p(y_i^s | x_i^s)$ is the predicted probability that the source image x_i^s belongs to identity y_i^s. $p(y_i^s | x_i^s)$ is obtained by the identity classifier of the source branch.

It is reported that the model trained using the labeled source data produces a high accuracy on the same distributed testing set. However, without adapting the model, the performance will deteriorate significantly when tested on an unseen (target) testing set. This deterioration is mainly caused by the domain shift and will be more serious with the increase of domain shift. Next, we will introduce the exemplar memory based target invariance learning method to improve the transferability of the model.

3.3 Invariance Learning for Target Domain

The deep re-ID model trained with only the source data is usually sensitive to the intra-domain variations of the target domain. In fact, the variations are critical factors influencing the performance in target testing set. Therefore, it is important and necessary to consider the intra-domain variations of the target domain during transferring the knowledge from the source domain to the target domain. To this end, this work investigates three underlying properties of target invariance and adapts the network with the constraints of the three properties. The three properties of target invariance are Exemplar-Invariance (EI), Camera-Invariance (CI) and Neighborhood-Invariance (NI), which are described as follows.

3.3.1 Three Properties of Target Invariance

Exemplar-invariance. Given a well-trained re-ID model, the top-ranked results are usually visually similar to the query. This suggests that the model has learned to discriminate persons by apparent similarity. However, this phenomenon may no longer apply when tested on a different distributed dataset. In fact, the appearance of each person image may be very different from others, even if they share the same identity. We call this property as exemplar-invariance, where each person image should be close to itself while far away from others. Therefore, by enforcing the exemplar-invariance on the target images, it is possible to enable the model to improve the ability of distinguishing person images based on the apparent representation.

Camera-invariance. Camera style variation is a natural and important factor in re-ID, where a person image may encounter significant changes in appearance under different cameras. A model trained using labeled source data can learn the camera-invariance for the source cameras, but may suffer from the variations caused by the target cameras. This is because the camera settings of the two domains will be very different. Inspired by HHL [64], we achieve the property of target camera-invariance by learning with unlabeled target images and their camera style transferred counterparts. These counterparts are in the style of different cameras but share the same identity with the original image.

Fig. 3. Example of camera style-transferred images on DukeMTMC-reID. A real image collected from a certain camera is transferred to images in the styles of other cameras. In this process, the identity information is preserved to some extent. The real image and its corresponding fake images are assumed to belong to the same class during camera-invariance learning.

In the constraint of camera-invariance, a target image and its camera-style transferred counterparts are encouraged to be close to each other. Suppose we have C cameras in the target set, we train CamStyle model [66] for the target domain with CycleGAN [68] or StarGAN [6]. For Market-1501 [58] and DukeMTMC-reID [61] that have less cameras, we use CycleGAN to train CamStyle model. For MSMT17 that has many cameras, we apply StarGAN to train CamStyle model. With the learned CamStyle model, each real target image collected from camera c is augmented with $C - 1$ images in the styles of other cameras while remaining the original identity. Examples of real images and fake images generated by the CamStyle model are shown in Fig. 3.

Neighborhood-invariance. Besides the camera-invariance property that is natural and can be easily identified, there are many crucial intra-domain variations that are hard to overcome without the annotation of identity, such as the variations of pose and background. In fact, for each target image, there may exist a number of positive samples in the target dataset. If we could exploit these positive samples during the adapting process, we are able to further improve the robustness of re-ID model in overcoming the variations in the target domain. Assuming that we are given an appropriate transferred model and a query target sample. The nearest neighbors of the query in the target set are most likely to be positive samples of the query. In light of this, we introduce the neighborhood-invariance under the assumption that a target image and its reliable neighbors will share the same identity and should be close to each other.

Next, we will introduce the loss functions of enforcing the three invariance constraints into the network.

3.3.2 Target Invariance Learning with Memory

An essential step of enforcing the three properties of invariance is to estimate the similarities (relationships) among target samples. Intuitively, a straightforward solution is to calculate the similarities between samples within a training mini-batch and enforce the three constraints by contrastive loss [18] or triplet loss [22]. However, the number of samples in a training mini-batch is far less than that in the entire

training set due to the limited GPU memory. This will cause two problems during invariance learning. First, it is hard to compose an appropriate mini-batch that includes various samples, and their corresponding CamStyle counterparts and neighbors. Second, the similarities between mini-batch samples are local and limited compared to the overall dataset. To address these problems, we introduce an exemplar memory into the network to store the up-to-date representations of the entire target samples. The memory enables the network to directly estimate the similarities between a training target sample and whole target samples, thereby effectively implementing the invariance constraints globally.

Exemplar Memory. The exemplar memory is a feature bank [52] that stores the up-to-date features of the entire dataset. Given an unlabeled target data including N_t images, we construct an exemplar memory ($\mathcal{F}^t$) which has N_t slots. Each slot stores the L2-normalized output of the feature extractor for the corresponding target image. In the initialization, we initialize the values of all the features in the memory to zeros. During each training iteration, for a target training sample x_i^t, we forward it through the model and obtain the L2-normalized output of the feature extractor, $f(x_i^t)$. During the back-propagation, we update the feature in the memory for the training sample x_i^t through,

$$\mathcal{F}^t[i] \leftarrow \alpha \mathcal{F}^t[i] + (1-\alpha)f(x_i^t), \tag{2}$$

where $\mathcal{F}^t[i]$ is the feature of image x_i^t in the *i-th* slot. The hyper-parameter $\alpha \in [0,1]$ controls the updating rate. $\mathcal{F}^t[i]$ is then L2-normalized via $\mathcal{F}^t[i] \leftarrow \|\mathcal{F}^t[i]\|_2$. Next, we will introduce the approach of memory based invariance learning.

Exemplar-invariance learning. The exemplar-invariance enforces a target image to be close to itself while far away from others. To achieve this goal, we regard the N_t target images as N_t different classes and apply a non-parameterized manner to classify each image into its own class. For simplicity, we assign the corresponding index as the class of each target sample. Specifically, given a target image x_i^t, we first compute the cosine similarities between the embedding $f(x_i^t)$ and features saved in the memory $\mathcal{F}^t$. Then, the predicted probability that x_i^t belongs to class i is calculated using softmax function,

$$p(i|x_i^t) = \frac{\exp(\mathcal{F}^t[i]^\mathrm{T} f(x_i^t)/\beta)}{\sum_{j=1}^{N_t} \exp(\mathcal{F}^t[j]^\mathrm{T} f(x_i^t)/\beta)}, \tag{3}$$

where $\beta \in (0,1]$ is temperature fact that balances the scale of distribution. The objective of exemplar-invariance learning is to minimize the negative log-likelihood over the target image x_i^t, as

$$\mathcal{L}_{ei} = -\log p(i|x_i^t). \tag{4}$$

Camera-invariance learning. The camera-invariance enforces a target image to be close to its style-transferred counterparts. To introduce the camera-invariance learning into the model, we try to classify each real image and its style-transferred counterparts to the same class. The loss function of camera-invariance is explained as,

$$\mathcal{L}_{ci} = -\log p(i|\hat{x}_i^t), \tag{5}$$

where $\hat{x}_i^t$ is a target image randomly sampled from the style-transferred images of x_i^t. In this way, images in different camera styles of the same sample are forced to be close to each other.

Neighborhood-invariance learning. The neighborhood-invariance enforces a target image to be close to its reliable neighbors. To endow this property into the network, we classify the target image x_i^t into the classes of its reliable neighbors. Supposing the selected reliable neighbors of x_i^t in the memory are denoted as $\mathcal{K}(x_i^t)$. We assign the weight of the probability that $x_{t,i}$ belongs to the class j as,

$$w_{i,j} = \begin{cases} \frac{1}{|\mathcal{K}(x_i^t)|}, & j \neq i \\ 1, & j = i \end{cases}, \forall j \in \mathcal{K}(x_i^t), \tag{6}$$

where $|\mathcal{K}(x_i^t)|$ denotes the size of $\mathcal{K}(x_i^t)$. The objective of neighborhood-invariance learning is formulated as a soft-label loss,

$$\mathcal{L}_{ni} = -\sum_{j \neq i} w_{i,j} \log p(j|x_i^t), \quad \forall j \in \mathcal{K}(x_i^t). \tag{7}$$

Note that, in order to distinguish between exemplar-invariance learning and neighborhood-invariance learning, x_i^t is not classified to its own class in Eq. 7.

The key of neighborhood-invariance learning is including as many positive samples as possible in $\mathcal{K}(x_i^t)$ while rejecting negative samples. In Section 3.4, we will introduce a vanilla neighbor selection (VNS) method and a graph-based positive prediction (GPP) method for exploring reliable neighbors $\mathcal{K}(x_i^t)$.

Overall loss of invariance learning. By jointly considering the exemplar-invariance, camera-invariance and neighborhood-invariance, the overall loss of invariance learning over a target training image can be written as,

$$\mathcal{L}_{tgt} = -\sum_j w_{i,j} \log p(j|\widetilde{x}_i^t), j \in \mathcal{K}(\widetilde{x}_i^t), \tag{8}$$

where $\widetilde{x}_i^t$ is an image randomly sampled from the union set of x_i^t and its camera style-transferred images. In Eq. 8, when $i = j$, we optimize the network with the exemplar-invariance learning and camera-invariance learning by classifying $\widetilde{x}_i^t$ into its own class. When $i \neq j$, the network is optimized with the neighborhood-invariance learning by leading $\widetilde{x}_i^t$ to be close to its reliable neighbors in $\mathcal{K}(\widetilde{x}_i^t)$.

3.4 Graph-based Positive Prediction

Vanilla Neighbor Selection (VNS). A naive way to select neighbors of a target image x_i^t is based on the cosine similarities between $f(x_i^t)$ and features saved in the memory. In the vanilla neighbor selection (VNS) method, we directly select the top-k ranked samples from the memory as the $\mathcal{K}(x_i^t)$. However, in such simple approach, the similarities between $f(x_i^t)$ and features are estimated independently, neglecting the relationships among the features in the memory. As a consequence, the similarity estimation of hard positive and negative samples may not be accurate, especially when the model has poor performance on the target domain.

To overcome this problem, next, we will propose a novel neighbor selection method, called graph-based positive prediction (GPP). The network of GPP is constructed by graph convolutional networks (GCNs) [21] and positive classifier.
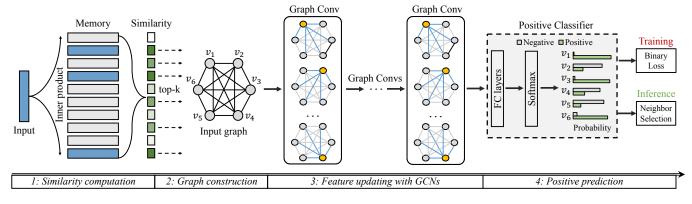
Fig. 4. The pipeline of graph-based positive prediction (GPP). Given the embedding of an input sample, 1) we first compute the similarities between the input and features in the memory. 2) The top-k ranked samples are selected as candidate neighbors and utilized to construct a graph for positive prediction. 3) The features of nodes are refined by graph convolutional networks (GCNs) on the graph. 4) The positive classifier is employed to predict positive probabilities of every node. For the source domain, the positive classification loss is computed for training the network of GPP. For the target domain, we select reliable neighbors for the input target sample, depending on the predicted positive probabilities of nodes.

The goal of GPP is to refine the similarities between $f(x_i^t)$ and features in the memory by leveraging the relationships among the features. To achieve this goal, the network of GPP is trained to predict probabilities of samples in the memory that belong to positive class of the input sample. We train the GPP network with labeled source samples and then apply it to infer reliable neighbors for target training samples.

3.4.1　Training GPP on Source Domain

To train the network of GPP, we simulate the positive prediction process on the source domain. In the implementation, we construct a source memory $\mathcal{F}^s$ with N_s slots for storing the features of source samples. The network of GPP includes several graph convolutional layers and a positive classifier. The training process of GPP is illustrated in Fig. 4, which is divided into four steps as described below.

1. Similarity computation. Given a training source sample x_i^s, we first extract its feature $f(x_i^s)$, and compute the cosine similarities between $f(x_i^s)$ and features in the source memory $\mathcal{F}^s$.

2. Graph construction. We select k-nearest-neighbors from the ranked-list as candidate neighbors of x_i^s, which is denoted as $V = \{v_1, v_2, ..., v_k\}$. Then, we construct a complete undirected graph $G(V, E)$, where V denotes the set of nodes and E indicates the set of edges. We denote $\mathcal{H} = \{\mathcal{F}^s[v_1], \mathcal{F}^s[v_2], ..., \mathcal{F}^s[v_k]\}$ as the node features. In order to encode the information of x_i^s, we normalize $\mathcal{H}$ by subtracting $f(x_i^s)$,

$$\mathcal{H} = \{\mathcal{F}^s[v_1] - f(x_i^s), ..., \mathcal{F}^s[v_k] - f(x_i^s)\}. \quad (9)$$

In practice, the $\mathcal{H}$ can be represented by a matrix with a size of $k \times d$, where d is the feature dimension of each node. As well, the weights of E are represented by an adjacency matrix $\mathcal{A} \in \mathbb{R}^{k \times k}$, where

$$\mathcal{A}_{i,j} = \mathcal{H}_i^T \mathcal{H}_j, \quad \forall i, j \in V. \quad (10)$$

Lastly, each row of $\mathcal{A}$ is normalized by softmax function.

3. Feature updating with GCNs. In this step, we aim to improve the node representations with the neighbor relations, so that can accurately predict positive samples from the candidates. To achieve this goal, we employ the graph convolution network (GCNs) [21], [46] to update

node features on $G(V, E)$. The input to the GCNs is a set of node features $\mathcal{H}$ and an adjacency matrix $\mathcal{A}$, and the output is a new set of node features $\mathcal{Z}$. Every graph convolutional layer in the GCNs can be written as a non-linear function,

$$\mathcal{H}^{(l+1)} = \text{ReLU}([\mathcal{A}\mathcal{H}^{(l)}||\mathcal{H}^{(l)}]\mathcal{W}^{(l)}), \quad (11)$$

with $\mathcal{H}^{(0)} = \mathcal{H}$ and $\mathcal{Z} = \mathcal{H}^{(L)}$. L is the number of graph convolutional layers. $||$ is the matrix concatenation operation. $\mathcal{W}^{(l)} \in \mathbb{R}^{2d_{in} \times d_{out}}$ is a learnable weight matrix for the l-th graph convolutional layer. d_{in} and d_{out} are the dimensions of input feature and out feature, respectively. In this paper, we adopt 4 graph convolutional layers to form the GCNs. The output $\mathcal{Z}$ is used to predict the positive probabilities for nodes by a positive classifier, as described in the next step.

4. Prediction with positive classifier. Given the updated features $\mathcal{Z}$, we use a positive classifier to predict the probability that a node belongs to the positive sample of the input x_i^s. For training of the GPP network, the loss function on the source domain is formulated by,

$$\mathcal{L}_{gpp} = -\frac{1}{k} \sum_j (y_j^* \log p_j^* + (1 - y_j^*) \log(1 - p_j^*)), \quad (12)$$

where $j \in V$. p_j^* is the predicted probability that node j belongs to the positive of the input x_i^s. y_i^* is the ground-truth binary label, defined as,

$$y_j^* = \begin{cases} 0, & y_j^s \neq y_i^s \\ 1, & y_j^s = y_i^s \end{cases}, \forall j \in V. \quad (13)$$

We optimize the network of GPP with $\mathcal{L}_{gpp}$ computed on the labeled source data.

3.4.2　Reliable Neighbors Selection on Target Domain

We infer the network of GPP on the target domain with the same steps as the training process on the source domain, except the loss computation in the last step. For a target sample x_i^t, we first compute the similarities between $f(x_i^t)$ and features in the target memory $\mathcal{F}^s$ with step 1. Then, we obtain the positive probabilities for the candidate neighbors V with the remaining steps. Finally, the reliable neighbors

Step 1   Step 2   Step 3   Step 4   Step 5   Step 6   Final
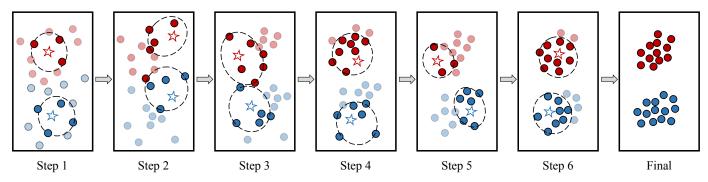
Fig. 5. Toy example of invariance learning. Dot colors denote classes. In each step, an input and its reliable neighbors (highlighted in circle) are enforced to be close by neighborhood-invariance learning, while an input and other samples (out of circle) are enforced to be far away by exemplar-invariance learning. With the interaction of exemplar-invariance and neighborhood-invariance, samples with the same class are gradually grouped closer, while dissimilar groups are separated.

$\mathcal{K}(x_i^t)$ are selected depending on the positive probabilities, defined as,

$$\mathcal{K}(x_i^t) = \{j | j \in V \wedge p_j^* \geq \mu\}, \qquad (14)$$

where μ is threshold value that controls whether a candidate should be selected as a reliable neighbor.

3.5 Final Loss for Network

By combining the losses of supervised learning, invariance learning and graph-based positive prediction learning, the final loss for the overall network is formulated as,

$$\mathcal{L} = \mathcal{L}_{src} + \mathcal{L}_{tgt} + \mathcal{L}_{gpp}. \qquad (15)$$

To this end, we introduce a framework for UDA in person re-ID, in which $\mathcal{L}_{src}$ aims to maintain a basic representation for person. Meanwhile, $\mathcal{L}_{tgt}$ attempts to take the knowledge from the labeled source domain and incorporate the invariance properties of the target domain into the network. L_{gpp} tries to learn a reliable neighbor prediction network that could facilitate the invariance learning. Note that, L_{gpp} is only calculated on the source domain and is only used to update the network of GPP.

3.6 Discussion on the Three Invariance Properties

We analyze the advantage and disadvantage for the three invariance properties, and the mutual effectiveness among them. The exemplar-invariance enforces each exemplar away from each other. It is beneficial to enlarge the distance between exemplars from different identities. However, exemplars of the same identity will also be far apart, which is harmful to the system. On the contrast, neighborhood-invariance encourages each exemplar and its neighbors to be close to each other. It is beneficial to reduce the distance between exemplars of the same identity. However, neighborhood-invariance might also pull closer images of different identities, because we could not guarantee that each neighbor shares the same identity with the input exemplar. Therefore, there exists a trade off between exemplar-invariance and neighborhood-invariance, where the former aims to lead the exemplars from different identities to be far away while the latter attempts to encourage exemplars of the same identity to be close to each other. In other words, the interaction process between exemplar-invariance

and neighborhood-invariance can be considered as a kind of local-clustering. On the one hand, when enforcing neighbor-invariance, samples with the same identity would be progressively grouped closer, through the neighborhood relation and the connection of their shared neighbors. On the other hand, exemplar-invariance encourages dissimilar samples to be pushed away from each other, so that dissimilar groups would be separated. Camera-invariance has the similar effect as the exemplar-invariance and also leads the exemplar and its camera-style transferred samples to share the same representation. A toy example of invariance learning is shown in Fig. 5.

4 EXPERIMENT

4.1 Dataset

We evaluate the proposed method on three large-scale person re-identification (re-ID) benchmarks: Market-1501 [58], DukeMTMC-reID [61] and MSMT17 [47].

Market-1501 [58] includes 32,668 labeled person images of 1,501 identities collected from six non-overlapping camera views. For evaluation, the dataset is divided into 12,936 images of 751 identities for training, 3,368 query images and 19,732 images of 705 identities for testing.

DukeMTMC-reID [61] is a re-ID benchmark collected from the DukeMTMC dataset [34]. The dataset is captured from eight cameras, including 36,411 person images from 1,812 identities. It contains 16,522 images of 702 identities for training, 2,228 query images of 702 identities and 17,611 gallery images for testing.

MSMT17 [47] is a newly released large-scale person re-ID benchmark. It is composed of 126,411 person images from 4,101 identities collected by an 15-camera system. The training set consists of 32,621 images of 1,041 identities, and the testing set contains 11,659 images as query and 82,161 images as gallery. The dataset has serious variations of scene and lighting, and is more challenging than the other two benchmarks.

Evaluation Protocol. During training, we use a labeled training dataset as the source domain and an unlabeled training dataset as the target domain. In testing, performance is evaluated on the target testing set by the cumulative matching characteristic (CMC) and mean Average Precision (mAP). For CMC, we use rank-1, rank-5, rank-10, and rank-20 as metrics.

4.2 Experimental Settings

Deep re-ID model. We adopt ResNet-50 [17] as the backbone of the feature extractor and initialize the model with the parameters pre-trained on ImageNet [8]. Specifically, we fix the first two residual layers and set the stride size of last residual layer to 1. After the global average pooling (GAP) layer, we add a batch normalization (BN) layer [20] followed by ReLU [31] and Dropout [39]. The identity classifier is an M-dimensional FC layer followed by softmax function. The input image is resized to 256×128. During training, we perform random flipping, random cropping and random erasing [63] for data augmentation. The probability of dropout is set to 0.5. We train the re-ID model with a learning rate of 0.01 for ResNet-50 base layers and of 0.1 for the others in the first 40 epochs. The learning rate is divided by 10 for the next 20 epochs. The SGD optimizer is used to train the re-ID model with a mini-batch size of 128 for both source and target domains. We initialize the updating rate of memory α to 0.01 and increase α linearly with the number of epochs, *i.e.*, $\alpha = 0.01 \times epoch$. Without specification, we set the temperature fact $\beta = 0.05$. For vanilla neighbor selection, we set $k = 8$ and directly select the top-k nearest-neighbors from the memory as the reliable neighbors. For GPP, we set the number of candidate neighbors $k = 100$ and neighbor selection threshold $\mu = 0.9$. We train the model with exemplar-invariance and camera-invariance learning at the first 10 epochs and add the neighborhood-invariance learning for the rest epochs. In testing, we extract the L2-normalized output of GAP as the image feature and adopt the Euclidean metric to measure the similarities between query and gallery images.

Network of graph-based positive prediction (GPP). The GPP network contains four graph convolutional layers and a positive classifier. The input and out dimensions of these graph convolutional layers are: 2048→2048, 2048→512, 512→256, 256→256. The positive classifier is composed of a 256-dimensional FC layer, a BN layer, a PReLU layer [16], and a 2-dimensional FC layer. We utilize SGD optimizer to update the GPP network after the 5*th* epoch. The learning rate is initialized as 0.01 and divided by 10 after the 40*th* epoch.

Baseline setting. We set the model as the *baseline* when trained the network using only the identity classifier. We employ the baseline in two ways: 1) *Train on target*, training the baseline on the labeled target training data and testing on the target testing set, and 2) *Source only*, training the baseline on a source labeled training set and directly applying to a target testing set without modification. The "Train on target" and the "Source only" can be considered as the upper bond and lower bond of our method, respectively.

4.3 Parameter Analysis

We first investigate the sensitivities of our approach to three important hyper-parameters, *i.e.*, the temperature fact β in Eq. 3, the number of candidate neighbors k in graph $G(V, E)$ and the neighbor selection threshold μ in Eq. 14. In order to clearly analyze the impact of every parameter, we vary the value of one parameter and keep fixed the others. Experiments are evaluated on the setting of transferring between Market-1501 and DukeMTMC-reID.

TABLE 1
Evaluation with different values of β in Eq. 3.

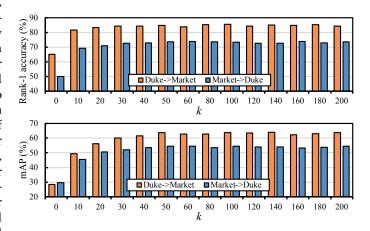| β	Duke → Market		Market → Duke	
	Rank-1	mAP	Rank-1	mAP
0.01	50.1	21.1	45.6	21.5
0.03	79.4	54.5	68.1	45.6
0.05	**84.1**	**63.8**	**74.0**	**54.4**
0.1	75.7	46.5	63.0	39.8
0.5	59.0	30.9	51.0	28.0
1.0	57.9	30.3	44.6	23.8

Fig. 6. Evaluation with different number of candidate samples for graph-based positive prediction.

Temperature fact β. In Table 1, we report the impact of the temperature fact β in Eq. 3. Assigning a lower value to β gives rise to a lower entropy, which commonly produces better results. However, when assigning a extremely low value to β, the model does not converge and produces a poor performance, *e.g.*, $\beta = 0.01$. The best results are obtained at $\beta = 0.05$.

Number of candidate neighbors k. In Fig. 6, we evaluate the performance of using a different number of candidate neighbors in the graph $G(V, E)$. When $k = 0$, our approach reduces to the model trained with only exemplar-invariance and camera-invariance. It is evident that injecting neighborhood-invariance into the network ($k > 0$) can significantly improve accuracy, and our method is insensitive to the changes of k. The rank-1 accuracy and mAP first improve with the increase of k and reach stable when $k > 100$. This improvement tendency is reasonable, because: 1) Using a larger k will include more positive samples in the graph and GPP might discover more hard positive samples for neighborhood-invariance learning; 2) Most positive samples are within the top-100 nearest neighbours, so it is unnecessary to include too many candidate samples. Considering the trade-off between accuracy and speed, we set $k = 100$ in our approach.

Threshold of positive neighbor selection μ. In Fig. 7, we compare the performance of using different values of μ in Eq. 14. On the one hand, assigning a too high value to μ would only select easy positive samples for neighbor-invariance learning, *e.g.*, $\mu = 0.99$. This will result in the model suffers from hard positive samples in testing. Note that, when $\mu = 1$, our method reduces to the model trained without neighbor-invariance learning. Since extremely few

TABLE 2
Methods comparison when transferred to Market-1501 and DukeMTMC-reID. **Train on target**: baseline model trained with the labeled target training data. **Source Only**: baseline model trained with only labeled source data. **EI**: exemplar-invariance. **CI**: camera-invariance. **NI**: neighborhood-invariance.

Methods	Invariance			DukeMTMC-reID → Market-1501					Market-1501 → DukeMTMC-reID				
	EI	CI	NI	R-1	R-5	R-10	R-20	mAP	R-1	R-5	R-10	R-20	mAP
Train on target	-	-	-	87.6	95.5	97.2	98.3	69.4	75.6	87.3	90.6	92.9	57.8
Source only	-	-	-	43.1	58.8	67.3	74.3	17.7	28.9	44.0	50.9	57.5	14.8
Ours	✓	-	-	48.7	67.4	74.0	80.2	21.0	34.2	51.3	58	64.2	18.7
Ours	✓	✓	-	63.1	79.1	84.6	89.1	28.4	53.9	70.8	76.1	80.7	29.7
Ours	✓	-	✓	71.8	83.1	87.1	90.6	45.7	67.2	80.0	83.8	86.7	48.3
Ours	✓	✓	✓	**84.1**	**92.8**	**95.4**	**96.9**	**63.8**	**74.0**	**83.7**	**87.4**	**90.0**	**54.4**

Fig. 7. Evaluation with different values of μ in Eq. 14

TABLE 3
Evaluation with different number of camera style samples for each target image. C is the number of cameras in the target domain. Model is trained with all three invariance constraints.
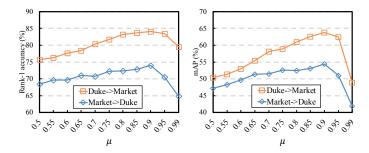
# CamStyle samples	Duke → Market		Market → Duke	
	Rank-1	mAP	Rank-1	mAP
0	71.8	45.7	67.2	48.3
1	83.5	61.1	72.1	52.3
3	84.0	62.8	73.0	53.0
C-1	**84.1**	**63.8**	**74.0**	**54.4**

samples would be selected as reliable positive neighbors. On the other hand, giving a too low value to μ might include too many false positive samples for neighbor-invariance learning, *e.g.*, $\mu < 0.7$. Approaching a sample to false positive samples would undoubtedly have deleterious effects on the results. Our approach reaches the best result when μ is around 0.9.

According to above analysis, we set $\beta = 0.05$, $k = 100$ and $\mu = 0.9$ in the following experiments.

4.4 Evaluation and Analysis

Performance of baseline in domain adaptation. In Table 2, we report the results of the baseline when transferring between Market-1501 and DukeMTMC-reID. When trained on the labeled target training set, the baseline (*Train on target*) achieves high accuracy. However, we observe a serious drop in performance when the baseline (*Source only*) is trained only using the labeled source set and directly applied to the target testing set. For the case of testing on Market-1501, the baseline (*Train on target*) achieves a rank-1 accuracy of 87.6%, but the rank-1 accuracy of the baseline (*Source only*) declines to 43.1%. A similar drop can be observed when transferred from Market-1501 to DukeMTMC-reID. This decline in accuracy is mainly caused by the domain shifts between datasets.

Ablation study on invariance learning. To study the effectiveness of the proposed invariance learning of target domain, we conduct ablation experiments in Table 2. We start from the basic model (Our method w/ EI), which enforces exemplar-invariance learning into the baseline (*Source only*) model, and then add camera-invariance, neighborhood-invariance, and both.

First, we show the effect of adding exemplar-invariance learning. As shown in Table 2, "Ours w/ EI" consistently improves the results over the baseline (*Source only*).

Specifically, the rank-1 accuracy improves from 43.1% to 48.7% and 28.9% to 34.2% when tested on Market-1501 and DukeMTMC-reID, respectively. This demonstrates that exemplar-invariance learning is an effective way to improve the discrimination of person descriptors for the target domain.

Second, we validate the effectiveness of camera-invariance learning over the basic model (ours w/ EI). In Table 2, we observe significant improvement when adding camera-invariance learning into the system. For example, "Ours w/ EI+CI" achieves a rank-1 accuracy of 63.1% when transferred from DukeMTMC-reID to Market-1501. This is higher than "Ours w/ EI" by 14.4% in rank-1 accuracy. The improvement demonstrates that the image variations caused by target cameras severely impact the performance in target testing set. Injecting camera-invariance learning into the model could effectively improve the robustness of the system to camera style variations. In Table 3, we also report the results of using different number of camera style samples for each target image. Even with only one camera style sample for each target image, the results of our approach can be considerable improved and are slightly lower than that of using all camera style samples. This indicates that the number of camera style samples in our approach can be agnostic to the number of cameras, and thus our approach is scalable to a scenario with many cameras.

Third, we evaluate the effect of neighborhood-invariance learning. As reported in Table 2, neighborhood-invariance significantly improves performance of the basic model (Ours w/ EI). When adding neighborhood-invariance to the basic model, "Ours w/ EI+NI" obtains 67.2% rank-1 accuracy and 48.3% mAP when transferred from Market-1501 to DukeMTMC-reID. This increases the results of the basic model by 33% in rank-1 accuracy and by 29.7% in mAP, respectively. A similar improvement is achieved when transferred to Market-1501.

Finally, we demonstrate the mutual benefit among the

TABLE 4
Results and computational cost analysis of the exemplar memory and graph-based positive prediction (GPP).

| Module | | Training time | GPU memory | Duke→ Market | | Market→ Duke | |
Memory	GPP	(s/iter)	(MB)	R-1	mAP	R-1	mAP
-	-	≈0.50	≈6,800	70.2	40.6	55.3	33.7
✓	-	≈0.52	≈7,000	77.4	45.5	65.4	42.6
✓	✓	≈0.63	≈9,800	84.1	63.8	74.0	54.4

TABLE 5
Results of using different neighbor selection methods. **VNS:** Vanilla neighbor selection. **GPP:** graph-based positive prediction.

| Neighbor selection | Condition | Duke→ Market | | Market→Duke | |
		R-1	mAP	R-1	mAP
VNS	top-8	77.4	45.5	65.4	42.6
Variant VNS	$p \geq 0.9$	77.0	46.4	64.5	42.8
Variant GPP	top-8	79.8	51.3	68.2	45.3
GPP	$p \geq 0.9$	**84.1**	**63.8**	**74.0**	**54.4**

Fig. 8. The curve of selected reliable neighbors in (a) recall and (b) precision throughout the training. Results are compared between vanilla neighbor selection (VNS) and graph-based positive prediction (GPP).

three invariance properties. As shown in the last row in Table 2, the three invariance properties are complementary to each other. The integration of them achieves obvious improvement over independently adding camera-invariance or neighborhood-invariance to the basic model. For example, "Ours w/ EI+CI+NI" achieves rank-1 accuracy of 84.1% when transferred from DukeMTMC-reID to Market-1501, outperforming "Ours w/ EI+CI" by 21% and "Ours w/ EI+NI" by 12.3%. Similar improvement is observed when transferred to DukeMTMC-reID. Particularly, our final model (Ours w/ EI+CI+NI) has only a small gap with the upper bond model (*Train on target*). For instance, our final model reaches rank-1 accuracy of 74.0% and mAP of 54.4% when transferred to DukeMTMC-reID, which is lower than "Train on target" by 1.6% in rank-1 accuracy and by 3.4% in mAP. This demonstrates that our approach has a strong capacity of bridging the gap between person re-ID domains.

Benefit of exemplar memory. In Table 4, we validate the effectiveness of the exemplar memory. When training our model without memory, we enforce the invariance learning within a mini-batch. Specifically, the inputs of the target mini-batch are composed of the target samples, corresponding CamStyle samples and corresponding k-nearest neighbors. Without the memory, we could not employ GPP in our model. Therefore, for a fair comparison, we train our memory based model without GPP, *i.e.* select the reliable neighbors $\mathcal{K}(x_i^t)$ by vanilla neighbor selection (VNS) method. As shown in Table 4, the exemplar memory based method clearly outperforms the mini-batch based method. This demonstrates the effectiveness of leveraging relations among whole datasets by memory. It is noteworthy that using the exemplar memory introduces limited additional training time (≈ + 0.02 s/iter) and GPU memory (≈ + 200 MB) compared to using the mini-batch.

Effectiveness of graph-based positive prediction. In Table 4, we report the results of training our method with and without graph-based positive prediction (GPP). Without GPP, our method reduces to training with vanilla neighbor selection (VNS), *i.e.* directly select k-nearest neighbors from memory as reliable neighbors $\mathcal{K}(x_i^t)$. It is clearly that, adding GPP into the system significantly improves the
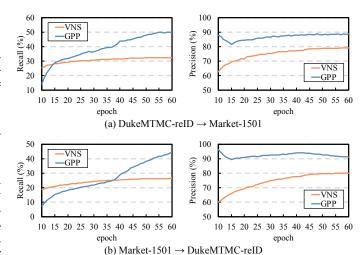
accuracy, requiring extra 0.11 s/iter training time and 2,800 MB GPU memory.

In Table 5, we also evaluate two other neighbor selection methods that are constructed as follows:

- Variant VNS: We train a positive prediction classifier as the same way as GPP, but without using graph convolution layers to update features of candidate samples. $\mathcal{K}(x_i^t)$ is selected according to the predicted positive probability and threshold μ. We set $\mu = 0.9$ for variant VNS.
- Variant GPP: We use the same architecture as the proposed GPP. The top-k samples are selected as $\mathcal{K}(x_i^t)$, according to the positive prediction of GPP. We set $k = 8$ for variant GPP.

Table 5 shows that: **(1)** GPP based methods consistently outperforms VNS based methods, whether selecting $\mathcal{K}(x_i^t)$ by top-k samples or threshold of predicted positive probability. This demonstrates the effectiveness of GPP, which updates features with the neighbor relations. **(2)** For GPP based methods, it is better to selecting reliable neighbors based on threshold than fix top-k samples. Because the number of positive samples of each sample is very different, using fix top-k samples might include negative samples that have low positive probability or ignore positive samples that have high positive probability. While threshold based strategy is more flexible and will always select samples with high positive probability. **(3)** The improvement of threshold based strategy is limited or even negative for the VNS method. Because the inputs of positive prediction classifier are unrelated features, which do not consider the relations

TABLE 6
Unsupervised person re-ID performance comparison with state-of-the-art methods on Market-1501 and DukeMTMC-reID. **Market**: Market-1501 [58]. **Duke**: DukeMTMC-reID [61]. **MSMT**: MSMT17 [47]. **SyRI**: SyRI [1]. **Multi**: a combination of seven datasets.

Methods	Reference	Market-1501					DukeMTMC-reID				
		Source	R-1	R-5	R-10	mAP	Source	R-1	R-5	R-10	mAP
LOMO [25]	CVPR 15	-	27.2	41.6	49.1	8.0	-	12.3	21.3	26.6	4.8
BOW [58]	ICCV 15	-	35.8	52.4	60.3	14.8	-	17.1	28.8	34.9	8.3
UMDL [33]	CVPR 16	Duke	34.5	52.6	59.6	12.4	Market	18.5	31.4	37.6	7.3
PTGAN [47]	CVPR 18	Duke	38.6	-	66.1	-	Market	27.4	-	50.7	-
PUL [10]	TOMM 18	Duke	45.5	60.7	66.7	20.5	Market	30.0	43.4	48.5	16.4
SPGAN [9]	CVPR 18	Duke	51.5	70.1	76.8	22.8	Market	41.1	56.6	63.0	22.3
CAMEL [55]	ICCV 17	Multi	54.5	73.1	-	26.3	Multi	40.3	57.6	-	19.8
MMFA [27]	BMVC 18	Duke	56.7	75.0	81.8	27.4	Market	45.3	59.8	66.3	24.7
SPGAN+LMP [9]	CVPR 18	Duke	57.7	75.8	82.4	26.7	Market	46.4	62.3	68.0	26.2
TJ-AIDL [45]	CVPR 18	Duke	58.2	74.8	81.1	26.5	Market	44.3	59.6	65.0	23.0
CamStyle [67]	TIP 19	Duke	58.8	78.2	84.3	27.4	Market	48.4	62.5	68.9	25.1
DECAMEL [56]	TPAMI 19	Multi	60.2	76.0	-	32.4	-	-	-	-	-
HHL [64]	ECCV 18	Duke	62.2	78.8	84.0	31.4	Market	46.9	61.0	66.7	27.2
DASy [1]	ECCV 18	SyRI	65.7	-	-	-	-	-	-	-	-
MAR [57]	CVPR 19	MSMT	67.7	81.9	-	40.0	MSMT	67.1	79.8	-	48.0
ECN [65]	CVPR 19	Duke	75.1	87.6	91.6	43.0	Market	63.3	75.8	80.4	40.4
Ours	This paper	Duke	**84.1**	**92.8**	**95.4**	**63.8**	Market	**74.0**	**83.7**	**87.4**	**54.4**

among candidates. In this case, the positive prediction classifier fails to refine the similarities between the input and candidates.

To further validate the effectiveness of GPP, we evaluate the selected reliable neighbors $\mathcal{K}(x_i^t)$ throughout the training for VNS and GPP. Two metrics are applied: recall and precision. Recall is the fraction of true positive samples in $\mathcal{K}(x_i^t)$ over the total amount of positive samples in the dataset. Precision is the fraction of true positive samples in $\mathcal{K}(x_i^t)$ among the number of samples in $\mathcal{K}(x_i^t)$. Fig. 8 shows that: 1) GPP consistently produces higher precision than VNS method. 2) The recall of GPP is lower than VNS in the early training epochs. However, the recall of GPP consistently grows with the training epochs and is largely higher than that of VNS in later training epochs. This confirms the superiority of GPP over VNS in terms of reliable neighbor selection and adaptation accuracy.

4.5 Comparison with State-of-the-art Methods

In Table 6 and Table 7, we compare our approach with state-of-the-art unsupervised learning methods when tested on Market-1501, DukeMTMC-reID and MSMT17.

SOTA on Market-1501 and DukeMTMC-reID. Table 6 reports the comparisons when tested on Market-1501 and DukeMTMC-reID. We compare with two hand-crafted feature based methods without transfer learning: LOMO [25] and BOW [58], four unsupervised methods that use a labeled source data to initialize the model but ignore the labeled source data during learning feature for the target domain: CAMEL [55], DECAMEL [56], UMDL [33] and PUL [10], and nine unsupervised domain adaptation approaches: PTGAN [47], SPGAN [9], MMFA [27], TJ-AIDL [45], DASy [1], MAR [57], CamStyle [67], HHL [64] and ECN [65].

We first compare with hand-crafted feature based methods, which neither require learning on labeled source set nor unlabeled target set. These two hand-crafted features have demonstrated the effectiveness on small datasets, but fail to produce competitive results on large-scale datasets. For example, the rank-1 accuracy of LOMO is 12.3% when

TABLE 7
Unsupervised/semi-supervised person re-ID performance comparison with state-of-the-art methods on MSMT17. *****: Using within-camera identity annotations of target dataset. **Multi**: a combination of seven datasets.

Methods	Reference	Source	MSMT17			
			R-1	R-5	R-10	mAP
TAUDL [22]	ECCV 18	MSMT*	28.4	-	-	12.5
UTAL [23]	TPAMI 19		31.4	-	-	13.1
DECA [56]	TPAMI 19	Multi	30.3	-	-	11.1
PTGAN [47]	CVPR 18		10.2	-	24.4	2.9
ECN [65]	CVPR 19	Market	25.3	36.3	42.1	8.5
Ours	This paper		**40.4**	**53.1**	**58.7**	**15.2**
PTGAN [47]	CVPR 18		11.8	-	27.4	3.3
ECN [65]	CVPR 19	Duke	30.2	41.5	46.8	10.2
Ours	This paper		**42.5**	**55.9**	**61.5**	**16.0**

tested on DukeMTMC-reID, which is much lower than transferring learning based methods.

Next, we compare with four unsupervised methods. Benefit from initializing model from the labeled source data and learning with unlabeled target data, the results of these three unsupervised approaches are commonly superior to hand-crafted methods. Such as, PUL obtains rank-1 accuracy of 45.5% when using DukeMTMC-reID as source set and tested on Market-1501, surpassing BOW by 9.7% in rank-1 accuracy.

Finally, we compare with the domain adaptation based approaches, which produce state-of-the-art results. As can be seen, our approach outperforms them by a large margin on both datasets. Specifically, our method achieves **rank-1 accuracy = 84.1%** and **mAP = 63.8%** when using DukeMTMC-reID as the source set and tested on Market-1501, and, obtains **rank-1 accuracy = 74.0%** and **mAP = 54.4%** vice-versa. It is worth noting that our approach significantly outperforms MAR [57] which uses a larger dataset (MSMT17) as the source domain. Compared to the current best method (ECN [65]), our method surpasses ECN by 20% and 14.4% in mAP when tested on Market-1501 and DukeMTMC-reID, respectively.

SOTA on MSMT17. We also evaluate our approach on a larger and more challenging dataset, *i.e.*, MSMT17.

Since it is a newly released dataset, there are only three unsupervised domain adaptation methods (PTGAN [47], DECA [57], and ECN [65]) reported results on MSMT17. In addition, we further compare with two semi-supervised methods, TAUDL [22] and UTAL [23], which use within-camera identity annotations of the target domain. As shown in Table 7, our approach clearly exceeds both unsupervised domain adaptation methods and semi-supervised methods. For instance, our method produces **rank-1 accuracy = 42.5%** and **mAP = 16.0%** when using DukeMTMC-reID as the source set, which is higher than ECN by 12.3% in rank-1 accuracy and by 5.8% in mAP. Similar superiority of our method can be observed when using the Market-1501 as the source domain.

5 CONCLUSION

In this paper, we propose an exemplar memory based unsupervised domain adaptation (UDA) framework for person re-ID task. With the exemplar memory, we can directly evaluate the relationships between target samples. And thus we could effectively enforce the underlying invariance constraints of the target domain into the network training process. Moreover, the memory enables us to design a graph-based positive prediction (GPP) method that can infer reliable neighbors from candidate neighbors. Experiment demonstrates the effectiveness of the invariance learning, memory and GPP for improving the transferable ability of deep re-ID model. Our approach produces new state of the art in UDA accuracy on three large-scale domains.

REFERENCES

[1] Slawomir Bak, Peter Carr, and Jean-Francois Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *Proc. ECCV*, 2018.

[2] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Proc. NIPS*, 2016.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *Proc. ICLR*, 2014.

[4] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proc. ICCV*, 2017.

[5] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Deep association learning for unsupervised video person re-identification. In *Proc. BMVC*, 2018.

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha2 Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proc. CVPR*, 2018.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009.

[9] 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 *Proc. CVPR*, 2018.

[10] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *ACM TOMM*, 2018.

[11] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[12] Douglas Gray and Hai Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Proc. ECCV*, 2008.

[13] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Proc. NIPS*, 2007.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. CVPR*, 2006.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. ICCV*, pages 1026–1034, 2015.

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

[18] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[19] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015.

[21] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2017.

[22] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 737–753, 2018.

[23] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised tracklet person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[24] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *Proc. CVPR*, 2018.

[25] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *Proc. CVPR*, 2015.

[26] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *Proc. ICLR*, 2014.

[27] Shan Lin, Haoliang Li, Chang-Tsun Li, and Alex Chichung Kot. Multi-task mid-level feature alignment network for unsupervised cross-dataset person re-identification. In *Prco. BMVC*, 2018.

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *Proc. ICML*, 2015.

[29] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proc. CVPR*, 2019.

[30] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[31] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proc. ICML*, 2010.

[32] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023, 2016.

[33] 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 *Proc. CVPR*, 2016.

[34] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Proc. ECCVW*, 2016.

[35] Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proc. ECCV*, 2018.

[36] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *Proc. ICML*, 2016.

[37] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. In *Proc. ECCV*, 2018.

[38] Kihyuk Sohn, Wenling Shang, Xiang Yu, and Manmohan Chandraker. Unsupervised domain adaptation for distance metric learning. In *Proc. ICLR*, 2019.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.

[40] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end

memory networks. In *Proc. NIPS*, 2015.

[41] Xiaoxiao Sun and Liang Zheng. Dissecting person re-identification from the viewpoint of viewpoint. In *Proc. CVPR*, 2019.

[42] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proc. ECCV*, 2018.

[43] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017.

[44] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Proc. NIPS*, 2016.

[45] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *Proc. CVPR*, 2018.

[46] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang. Linkage based face clustering via graph convolution network. In *Proc. CVPR*, 2019.

[47] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proc. CVPR*, 2018.

[48] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Proc. ICLR*, 2015.

[49] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019.

[50] Zhirong Wu, Alexei A Efros, and Stella X Yu. Improving generalization via scalable neighborhood component analysis. In *Proc. ECCV*, 2018.

[51] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. CVPR*, 2018.

[52] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proc. CVPR*, 2017.

[53] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, 2017.

[54] Fengxiang Yang, Zhun Zhong, Zhiming Luo, Sheng Lian, and Shaozi Li. Leveraging virtual and real person for unsupervised person re-identification. *arXiv preprint arXiv:1811.02074*, 2018.

[55] Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Cross-view asymmetric metric learning for unsupervised person re-identification. In *Proc. ICCV*, 2017.

[56] Hong-Xing Yu, Ancong Wu, and Wei-Shi Zheng. Unsupervised person re-identification by deep asymmetric metric embedding. *IEEE TPAMI*, 2019.

[57] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. In *Proc. CVPR*, 2019.

[58] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proc. ICCV*, 2015.

[59] Liang Zheng, Yi Yang, and Alexander G Hauptmann. Person re-identification: Past, present and future. *arXiv*, 2016.

[60] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *Proc. CVPR*, 2019.

[61] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proc. ICCV*, 2017.

[62] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *Proc. CVPR*, 2017.

[63] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv*, 2017.

[64] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero- and homogeneously. In *Proc. ECCV*, 2018.

[65] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identication. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019.

[66] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *Proc. CVPR*, 2018.

[67] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camstyle: A novel data augmentation method for person re-identification. *IEEE TIP*, 2019.

[68] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. ICCV*, 2017.