Unsupervised Person Re-Identification by Deep Asymmetric Metric Embedding

Hong-Xing Yu [ID], Ancong Wu [ID], and Wei-Shi Zheng [ID]

Abstract—Person re-identification (Re-ID) aims to match identities across non-overlapping camera views. Researchers have proposed many supervised Re-ID models which require quantities of cross-view pairwise labelled data. This limits their scalabilities to many applications where a large amount of data from multiple disjoint camera views is available but unlabelled. Although some unsupervised Re-ID models have been proposed to address the scalability problem, they often suffer from the view-specific bias problem which is caused by dramatic variances across different camera views, e.g., different illumination, viewpoints and occlusion. The dramatic variances induce specific feature distortions in different camera views, which can be very disturbing in finding cross-view discriminative information for Re-ID in the unsupervised scenarios, since no label information is available to help alleviate the bias. We propose to explicitly address this problem by learning an unsupervised asymmetric distance metric based on cross-view clustering. The asymmetric distance metric allows specific feature transformations for each camera view to tackle the specific feature distortions. We then design a novel unsupervised loss function to embed the asymmetric metric into a deep neural network, and therefore develop a novel unsupervised deep framework named the *DE*ep *C*lustering-based *A*symmetric *ME*tric *L*earning (*DECAMEL*). In such a way, DECAMEL jointly learns the feature representation and the unsupervised asymmetric metric. DECAMEL learns a compact cross-view cluster structure of Re-ID data, and thus help alleviate the view-specific bias and facilitate mining the potential cross-view discriminative information for unsupervised Re-ID. Extensive experiments on seven benchmark datasets whose sizes span several orders show the effectiveness of our framework.

Index Terms—Unsupervised person re-identification, unsupervised metric learning, unsupervised deep learning, cross-view clustering, deep clustering

✦

1 INTRODUCTION

ALONG with the extensive deployment of visual surveillance networks, considerable visual surveillance data is emerging everyday within the networks. A basic problem of analyzing and exploiting the data is to find target persons who had been previously observed like missing children and suspects, also known as person re-identification (Re-ID) [1], [2]. Typically, Re-ID focuses on pedestrian matching and ranking across multiple non-overlapping camera views.

Re-ID remains an open problem although it has received increasing explorations in recent years, principally because of dramatic intra-person appearance variation across views and high inter-person similarity. They mainly focus on learning robust and discriminative feature representations [3], [4], [5],

---

- *H.-X. Yu is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China, and also with the Guangdong Province Key Laboratory of Information Security, 510006, China. E-mail: xKoven@gmail.com.*
- *A. Wu is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510275, China, and also with the Collaborative Innovation Center of High Performance Computing, NUDT, China. E-mail: wuancong@mail2.sysu.edu.cn.*
- *W.-S. Zheng is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510275, China, and also with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, 510006, China. E-mail: wszheng@ieee.org.*

[6], [7], [8], [9], [10], [11], [12], [13] and distance metrics [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28]. Recently, deep learning has been adopted to the Re-ID community and has achieved promising performances [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40]. They have contributed a lot to the Re-ID community.

However, supervised models are intrinsically limited because they rely on a large amount of correctly labelled cross-view training data, which is very expensive [41]. In the context of Re-ID, this limitation is even pronounced because *(1)* manual labelling may not be fully reliable when a huge number of images to be checked across multiple camera views, and more importantly *(2)* the astronomical cost of time and money is prohibitive to label the overwhelming amount of data. In many scenarios, there is a large amount of data available but unlabelled, so that the supervised methods would be restricted or not applicable.

To directly make full use of the cheap and valuable unlabelled data, some efforts on unsupervised Re-ID have been made [42], [43], [44], [45], [46], [47], [48], [49], [50], which typically learn general/universal feature projections for person images from every camera view, but the performances are still not satisfactory. One of the main reasons is that without the supervision of cross-view pairwise labelled data, it is very difficult for a universal feature projection to capture the cross-view discriminative information under dramatic cross-view person appearance variations caused by view-specific conditions (Fig. 1a). For example, a person in white may
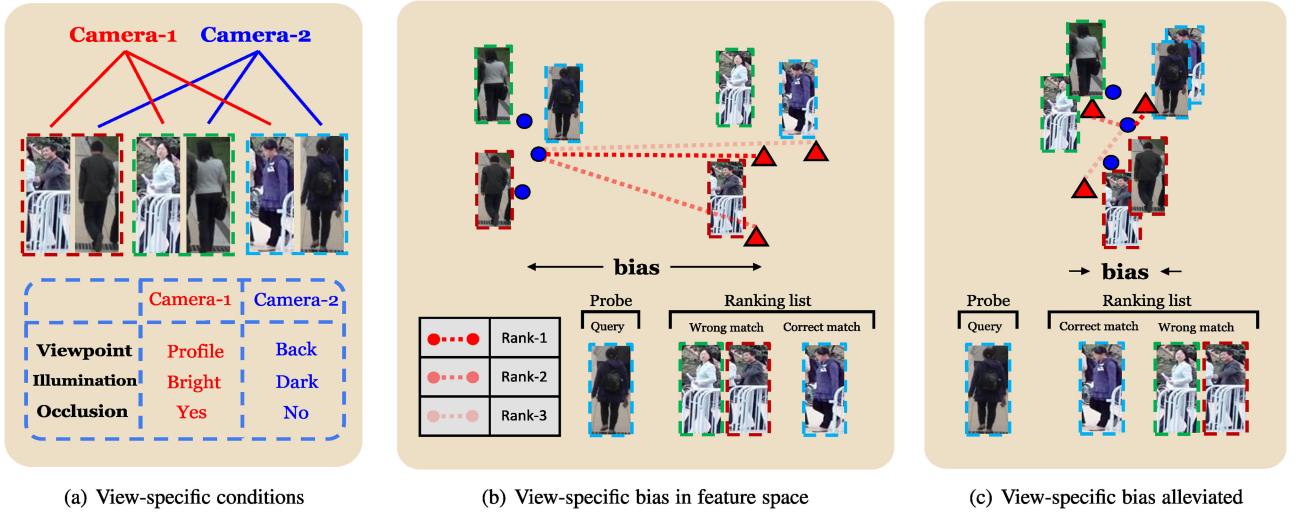
Fig. 1. Cause and effect of the view-specific bias. *(a)* The images from different camera views suffer from dramatic viewing condition variations across camera views. Images in the same colored box belong to the same person. *(b)* The specific viewing conditions lead to the view-specific feature distortions/bias, making unsupervised Re-ID more challenging. A visualization of the view-specific bias can be found in Fig. 3a. For example, given a probe image (in blue box), the correct match falls to the third rank due to the view-specific bias. *(c)* If we can tackle the view-specific feature distortion (i.e., alleviate the view-specific bias), we may reach a better cross-view matching performance. Further visualization and discussion can be found in Figs. 3 and 4. (Best viewed in color).

appear wearing gray in one camera view where illumination is darker, while he may appear snow-white in another view where illumination is brighter. Without the pairwise supervision guidance, it is very hard for a universal feature projection to map such drastically different cross-view image features of the same person to very close points in the subspace. More generally, the view-specific conditions introduce the view-specific bias, i.e., some specific feature distortions in different camera views, which can be very disturbing in finding what is more distinguishable in matching people across views. We show a toy example to illustrate this disturbing effect in Fig. 1. In Fig. 1a, the color feature of a person's arm may be located in the central position of images from Camera-1 since Camera-1 typically captures the profile of a person, while the corresponding color feature may appear at the boundary of images from Camera-2 since it captures the back of the person. Thus, the view-specific feature distortion can make the cross-view matching even harder as shown in Fig. 1b. In particular, most existing unsupervised models treat the samples from different views in the same manner, and thus could suffer from the effect of the view-specific bias.

In this work, we propose to *explicitly* deal with the view-specific bias problem in unsupervised Re-ID by formulating it as an unsupervised asymmetric distance metric learning problem. We briefly introduce the idea in the following. Given a pair of sample feature representations $\mathbf{x}_i$ and $\mathbf{x}_j$, a conventionally learned distance metric between them is:

$$d_l(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \boldsymbol{M} (\mathbf{x}_i - \mathbf{x}_j)} = \|\boldsymbol{U}^{\mathrm{T}}\mathbf{x}_i - \boldsymbol{U}^{\mathrm{T}}\mathbf{x}_j\|_2, \quad (1)$$

where $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}$ is a positive semi-defined matrix, and $\boldsymbol{U}$ is a transformation matrix. Learning such a metric is equivalent to finding a space "shared" by samples from each camera view [51]. This shared space is found by projecting all samples with a view-generic universal transformation $\boldsymbol{U}$. However, different camera views may induce different feature characteristics, e.g., the side-view of persons in Camera-1

versus the back-view in Camera-2 as shown in Fig. 1a. Intuitively, it is important to perform view-specific transformations for acquiring common features to match person images across camera views (e.g., selecting the corresponding color features at different locations of images from different camera views). Therefore, it can be hard for a universal transformation to implicitly tackle the view-specific feature distortions from different camera views, especially when we lack label information to guide it. This motivates us to explicitly take the view-specific feature distortion into account. Inspired by the supervised asymmetric distance model [3], [28], we propose to embed the asymmetric metric learning into our unsupervised Re-ID modelling, and thus consider the modification of Eq. (1), i.e., the asymmetric form:

$$d_l(\{\mathbf{x}_i, v_i\}, \{\mathbf{x}_j, v_j\}) = \|\boldsymbol{U}_{v_i}^{\mathrm{T}}\mathbf{x}_i - \boldsymbol{U}_{v_j}^{\mathrm{T}}\mathbf{x}_j\|_2, \quad (2)$$

where v_i denotes which camera view the ith sample comes from, and $\boldsymbol{U}_{v_i}$ is the corresponding view-specific transformation. Such an asymmetric metric allows specific transformation for each view to tackle the view-specific feature distortions.

Since no label information is provided to strictly distinguish every visually similar person in unsupervised Re-ID scenarios, we encourage the asymmetric metric to condense the visually similar cross-view person image *clusters*, and thus better distinguish them from other dissimilar clusters. With the asymmetric metric in the clustering procedure, we can explicitly address the view-specific bias and learn a better cross-view cluster structure of the Re-ID data in the shared space.

In the following, we refer to the clustering procedure which uses an asymmetric metric as the *asymmetric metric clustering*. Based on the asymmetric metric clustering, we first develop a linear metric learning model named Clustering-based *Asymmetric MEtric Learning* (*CAMEL*). CAMEL jointly learns an asymmetric metric and a cluster separation. Then, based on CAMEL, we further propose a novel unsupervised deep framework named *DEep CAMEL* (*DECAMEL*), which

jointly learns the feature representation and the unsupervised asymmetric metric end to end. DECAMEL can address the sub-optimality due to the separation of feature and metric learning. By learning a better cross-view cluster structure in the shared space, DECAMEL attempts to mine the underlying cross-view discriminative information to achieve a better cross-view matching performance.

More specifically, DECAMEL consists of a feature extractor network and an asymmetric metric layer. We propose a novel unsupervised loss function for the whole deep framework. In the optimization of DECAMEL, the asymmetric metric layer is initialized by CAMEL, which can alleviate the view-specific bias and learn a preliminary cross-view cluster separation. Then, the asymmetric metric is embedded into the whole network by the joint learning. The term "embedded" refers to the fact that, during the joint learning, the asymmetric metric is back-propagated to the whole network, so that finally the view-specific bias in the feature space is also alleviated. This is empirically observed in Fig. 4. Through this joint learning procedure, DECAMEL learns a compact cross-view cluster structure of Re-ID data. And thereby, it attempts to mine the potential cross-view discriminative information which will be qualitatively illustrated in Section 4 and quantitatively validated in Section 5.

In summary, our *main contributions* in this work are:

1) We formulate unsupervised Re-ID as a joint learning problem which consists of learning the feature representation and an asymmetric distance metric, together with a cluster separation. To our best knowledge, this is the first work formulating unsupervised Re-ID as a joint learning problem of the feature and the metric.

2) We propose a novel unsupervised deep framework named DECAMEL. Different from previous works in unsupervised Re-ID, DECAMEL jointly learns the feature representation and the unsupervised asymmetric metric. DECAMEL can learn a compact cross-view clustering structure for mining underlying cross-view discriminative information. We also propose a novel unsupervised asymmetric metric learning model, i.e., CAMEL, for the metric initialization in DECAMEL. CAMEL allows to explicitly model the view-specific conditions in unsupervised Re-ID [52], and thus the useful information of the cross-view person appearance variations can be exploited for the feature learning in DECAMEL.

3) For large-scale view-extendable scenarios, we develop a method named View Clustering (VC) for better generalizability and scalability. VC provides a flexible control on the generalizability versus ability to precisely model the view-specific conditions. To evaluate our framework, we conduct extensive experiments on seven size-varying benchmark datasets. Experimental results show that our model outperforms the state-of-the-art with noticeable margins, indicating that the asymmetric modelling is effective in unsupervised Re-ID.

2 RELATED WORK

2.1 Unsupervised Re-ID Models

While there are a lot of great works in supervised Re-ID [3-40], unsupervised Re-ID still remains under-studied. Existing attempts in unsupervised Re-ID can be classified into two categories: feature representation learning [42], [43], [44], [45], [46], [47] and dictionary learning [48], [49], [50].

Feature Representation Learning. This category of models mainly focuses on designing or learning discriminative and invariant features. Farenzena et al. [42] proposed to extract features containing three complementary parts of human appearance, including color, spatial arrangement of colors and texture patches. Cheng et al. [43] proposed to exploit the part-to-part correspondence. They evaluate the modified HSV characterization for each part and found the maximally stable color regions. Zhao et al. [44] proposed to exploit salience information by building dense correspondence and unsupervised salience learning. Wang et al. [45] proposed to localize person foreground saliency and remove busy background clutters. Lisanti et al. [46] proposed to extract the weighted histogram of overlapping stripes (WHOS) feature, and then applied the Iterative Re-Weighted Sparse Ranking (ISR) algorithm to generate the ranked list of gallery individuals. Wang et al. [47] proposed a CCA-based model to learn a feature subspace where within-view similar persons are distant from each other and cross-view similar persons are close to each other. Recently, Fan et al. [53] attempted to learn feature representation with a deep clustering network.

Dictionary Learning. Dictionary learning aims to learn a dictionary with its atoms corresponding to some semantic elements. The learned dictionary will be used to produce the new features which minimize the reconstruction error. Kodirov et al. [48] proposed to formulate unsupervised Re-ID as a sparse dictionary learning problem. To regularize the learned dictionary, they learned it with graph Laplacian regularization, and iteratively updated the graph Laplacian matrix. Then they took a further step [50]. They propose to introduce an l_1-norm graph Laplacian to jointly learn the graph and the dictionary, resulting in alleviation of the effects of data outliers [50]. Peng et al. [49] proposed to learn a dictionary by unsupervised multi-task learning. Different from [48] and [50], the learned dictionary consists of task-shared, task-specific and unique components.

Our model is very different from them in that: *(1)* Ours explicitly addresses the view-specific bias by learning an asymmetric metric, i.e., learning view-specific transformation for each camera view. Note that the RKSL [47] is based on Canonical Correlation Analysis (CCA) and also learns different projections for different views. However, it does not address the view-specific bias problem, because the CCA-based model RKSL learns two projections separately and inconsistently. RKSL focuses on the correlations of cross-view samples by maximizing the correlation coefficients. In the context of unsupervised learning (given no cross-view labelled pairs), not like ours, RKSL is not able to discover the cross-view cluster structure and thus not suitable to mine the potential discriminative information across views. In addition, no inherent consistency between two specific feature projections is considered, while our model preserves cross-view consistency by learning an asymmetric metric under a cross-view consistency regularization. *(2)* Ours can jointly learn the feature representation and the asymmetric metric end to end.

This work is based on our preliminary work which presented CAMEL [52]. In addition to giving a more detailed description and analysis of the proposed linear model CAMEL, the major differences are as follow: *(1)* We present a novel unsupervised deep framework DECAMEL. Compared

to the linear metric learning model CAMEL, we propose the novel loss function, architecture and learning algorithm, which together enable DECAMEL to perform joint learning of feature representation and metric in an unsupervised way. By this means, compared to CAMEL, DECAMEL can learn a more compact cross-view cluster structure, potentially facilitating mining the underlying cross-view discriminative information. (2) We propose the view clustering that can significantly improve the generalizability and scalability of DECAMEL. (3) We present more in-depth discussion and analysis on the proposed framework, including a series of visual results which show intuitively and progressively how DECAMEL works. Moreover, we conduct more extensive evaluations for comparisons and analysis.

2.2 Unsupervised Metric Learning

Although unsupervised metric learning has not been exploited in Re-ID, there are a few works that exploit unsupervised metric learning in other fields [54], [55], [56], [57]. Ye et al. [54] proposed to jointly learn a metric and a clustering separation for better clustering results. Qin et al. [55] proposed to learn a metric based on regularized neighborhood component analysis for clustering analysis. Cinbis et al. [56] proposed to mine video information to find positive and negative pairs of faces, and thus learning a metric for video face recognition. Jiang et al. [57] proposed a diffusion-based approach to improve an input similarity metric for vision tasks like image segmentation and clustering.

Specifically, our model is closely related to the Adaptive Metric Learning (AML) [54]. AML learns a transformation G to project the training data $\mathbf{x}$ onto a low dimensional space: $\hat{\mathbf{x}} = G^{\mathrm{T}}\mathbf{x}$. It minimizes the sum of squared error (SSE) in the subspace:

$$SSE = \sum_{\hat{\mathbf{x}}} d_M(\hat{\mathbf{x}}, \hat{\mathbf{c}})^2, \tag{3}$$

where d_M is the Mahalanobis distance, and $\hat{\mathbf{c}}$ denotes the cluster centroid to which $\hat{\mathbf{x}}$ belongs.

Our model is different from AML in that ours performs cross-view clustering and explicitly models the view-specific bias in the context of unsupervised Re-ID. We propose to alleviate the bias by learning an asymmetric metric. Furthermore, we propose to embed the metric learning into the deep neural networks for a joint learning of feature representation and the asymmetric metric.

The asymmetric modelling is also used in cross-modal retrieval [58], [59], [60], where Canonical Correlation Analysis [61] and Partial Least Squares (PLS) [62] based methods are arguably the most popular. CCA and PLS also learn different projections for different modalities to induce a latent space, where the correlation/covariance of pairwise cross-modal training samples is maximized. However, they do not address the view-specific bias problem in person Re-ID since the projections are learned separately and inconsistently. Moreover, they require sufficient pairwise labelled data is required by these methods. There are also some works on supervised deep metric learning [63], [64], [65], [66], [67], [68]. However, these models require substantial labelled data for training. In contrast, our model can directly learn from unlabelled data and thus it is free from requiring large amount of expensive labelled data.

2.3 Unsupervised Deep Clustering Embedding

Our model is also related to the unsupervised deep clustering embedding technique [69], [70], [71]. Xie et al. [69] proposed to jointly learn cluster membership and deep representation by minimizing the KL divergence between original data distribution and a target distribution. Yang et al. [70] proposed to jointly learn image clusters and deep representation using a convolutional neural network. Wang et al. [71] proposed to learn a task-specific deep architecture based on sparse coding for clustering.

DECAMEL is different from all these models in that it joins unsupervised asymmetric metric learning with cross-view clustering, which is specifically designed for cross-view matching in Re-ID, while others use standard symmetric metrics in clustering.

3 APPROACH

In this section, we progressively develop our unsupervised Re-ID framework, i.e., the *DE*ep Clustering-based *A*symmetric *ME*tric *L*earning (*DECAMEL*). Specifically, our framework first learns an initial asymmetric metric by a linear unsupervised model (CAMEL) and then embeds the metric into a deep network by jointly learning feature and metric. These two steps are based on the asymmetric metric clustering, so we first introduce the asymmetric metric clustering and develop the linear model. Then, we introduce a novel loss function for the unsupervised deep joint learning. We show an overview of our framework in Fig. 2.

3.1 Unsupervised Asymmetric Metric Learning

Now let us dive into the details and develop the model step by step. Under a general unsupervised Re-ID setting, we have V camera views. From each of them, we have collected N_v ($v = 1, \ldots, V$) images, and thus we have $N = \sum_{v=1}^{V} N_v$ unlabelled images for training. Here we assume that feature representation for the training images are given, and the training set is denoted as $\mathcal{X} = \{\mathbf{x}_i, v_i\}_{i=1}^{N}$, where $\mathbf{x}_i$ is the feature vector and v_i denotes which camera view it comes from. Note that in visual surveillance, the camera view "label" v_i is naturally available for each image, since its straightforward to know by which camera an image is captured in a camera network. Here we follow a popular assumption that during training and testing, the camera views are the same [3, 5, 6, 8, 9, 11, 14, 15, 18-40, 44, 47-50]. We will also discuss a novel view-extendable setting in Section 3.4.

We are looking for some transformations to map the data into a shared space, where we can better separate the images of different persons. A natural idea is to decrease the intra-class (here, a class is a person) discrepancy and enlarge the inter-class discrepancy. In an unsupervised scenario, however, we have no labelled data to strictly separate visually similar persons. Therefore, we relax the original idea: we focus on gathering similar person images together, and thereby separating relatively dissimilar ones. We formulate it by an objective like that of k-means clustering [72]:

$$\min_{U, \{\mathbf{c}_k\}_{k=1}^{K}} f_{intra} = \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \|U^{\mathrm{T}}\mathbf{x}_i - \mathbf{c}_k\|^2, \tag{4}$$
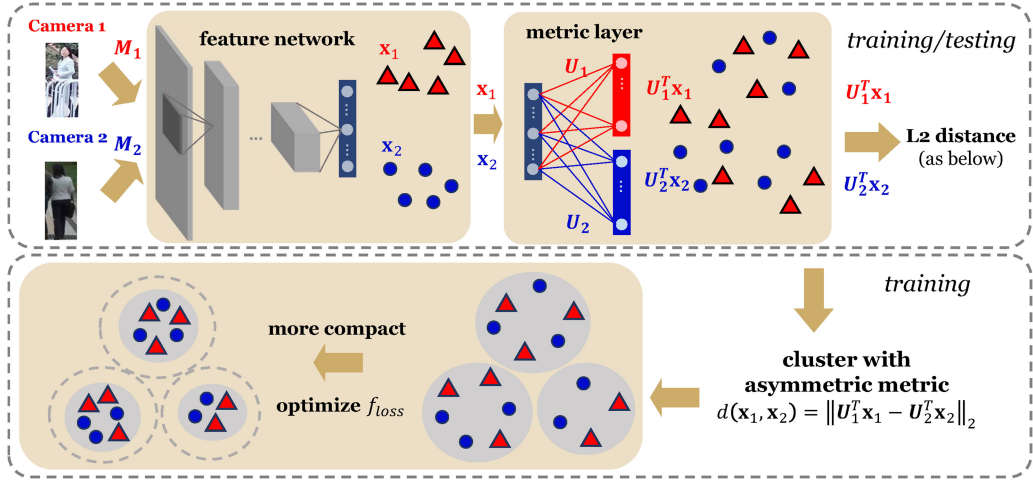
Fig. 2. Illustration of our framework DECAMEL. We follow the brown arrows to inspect it. We extract features for person images by a deep network. Due to view-specific conditions, the initial feature space has severe view-specific bias: the red triangles (data points from Camera 1) and blue circles (from Camera 2) are far apart. We perform CAMEL to learn an initial asymmetric metric. In the shared space produced by the asymmetric metric, the view-specific bias is alleviated. By optimizing the proposed unsupervised loss, DECAMEL jointly learns the feature representation and asymmetric metric in an end-to-end manner. During testing, pairwise distance can be computed by Eq. (2). (Best viewed in color).

where U is the view-generic universal feature transformation, K is the number of clusters, $\mathbf{c}_k$ denotes the centroid of the kth cluster and $\mathcal{C}_k = \{i|U^{\mathrm{T}}\mathbf{x}_i \in k\text{th cluster}\}$.

However, clustering can be largely affected by the view-specific bias when applied in the cross-view problems. In Re-ID, the variances across camera views like different lighting conditions, human pose variations and occlusions [28] can be very dramatic. They are disturbing or even dominating in searching the similar person images across views during clustering procedure. To address this problem, we learn specific transformations for each view rather than a generic one, to explicitly take the view-specific feature distortion into account and to alleviate the view-specific bias. As discussed in Eq. (2), the idea can be further formulated by

$$\min_{\{U_v\}_{v=1}^V,\{\mathbf{c}_k\}_{k=1}^K} f_{intra} = \frac{1}{N}\sum_{k=1}^K\sum_{i\in\mathcal{C}_k}\|U_{v_i}^{\mathrm{T}}\mathbf{x}_i - \mathbf{c}_k\|^2 \quad (5)$$
$$s.t. \quad U_v^{\mathrm{T}}\mathbf{\Sigma}_v U_v = I \quad (v=1,\dots,V),$$

where U_{v_i} is the specific feature transformation for the v_ith camera view, v_i denotes which camera view $\mathbf{x}_i$ comes from, $\mathbf{\Sigma}_v = \sum_{\mathbf{x}_t:v_t=v}\mathbf{x}_t\mathbf{x}_t^{\mathrm{T}}/N_v + I$ is a covariance matrix, and I represents the identity matrix which is used to avoid singularity of the covariance matrix. The transformation for each instance $\mathbf{x}_i$ is determined by v_i. The quasi-orthogonal constraints on U_v ensure that the model will not simply give zero matrices. By jointly learning the asymmetric metric and cross-view clustering, we actually realize an asymmetric metric clustering on Re-ID data across camera views.

Mathematically, if we minimize this objective function, every U_v will largely depend on the data distribution from the vth view. Since there is view-specific bias on each view, any pair of transformations—U_v and U_w—could be arbitrarily different according to the biases. However, large inconsistencies among the learned transformations are not what we expect, since these transformations are with respect to person images from different views. Although under different conditions, the subjects are human beings, and thus they are inherently correlated and not heterogeneous. Therefore, largely

different projection basis pairs would fail to capture the discriminative nature of the person images.

Hence, to strike a balance between the ability to preserve the cross-view consistency and the ability to alleviate view-specific bias, we add a cross-view consistency regularization term to our objective function. The cross-view consistency regularization penalizes the discrepancy between any pair of correlated transformation basis $\mathbf{u}_v^c$ and $\mathbf{u}_w^c$, where $\mathbf{u}_v^c$ is the cth column of U_v. Thus, we formulate it as:

$$f_{consistency} = \sum_{v,w}\sum_c\|\mathbf{u}_v^c - \mathbf{u}_w^c\|_2^2 = \sum_{v,w}\|U_v - U_w\|_F^2, \quad (6)$$

where $\|\cdot\|$ is the Frobenius norm of a matrix. And then, our optimization task is given by

$$\min_{\{U_v\}_{v=1}^V,\{\mathbf{c}_k\}_{k=1}^K} f_{obj} = f_{intra} + \lambda f_{consistency}$$
$$= \frac{1}{N}\sum_{k=1}^K\sum_{i\in\mathcal{C}_k}\|U_{v_i}^{\mathrm{T}}\mathbf{x}_i - \mathbf{c}_k\|^2 + \lambda\sum_{v,w}\|U_v - U_w\|_F^2$$
$$s.t. \quad U_v^{\mathrm{T}}\mathbf{\Sigma}_v U_v = I \quad (v=1,\dots,V),$$
$$\quad (7)$$

where λ is the cross-view regularizer. We call the above model the **C**lustering-based **A**symmetric **ME**tric **L**earning (*CAMEL*).

We will show by an illustration that the asymmetric metric can alleviate the view-specific bias in the Re-ID problem in Section 4.1, and show that the cross-view consistency regularization contributes much in our framework in Section 5.4.2.

Remark 1. (Cross-view Consistency Regularization for the Metric). We note that although asymmetric metric learning has been successfully applied in supervised Re-ID [3], [28], it is a pseudo metric rather than a strict metric [28], because it may not meet the coincidence property: given two identical feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\mathbf{x}_i = \mathbf{x}_j$) from different camera views v_i and v_j, the asymmetric metric may not guarantee $d(\mathbf{x}_i, \mathbf{x}_j) = 0$. In this aspect, the cross-view consistency regularization plays a role to control an upper

bound of coincidence discrepancy. In fact, according to the Cauchy Inequality, we have

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|U_{v_i}^\mathrm{T}\mathbf{x}_i - U_{v_j}^\mathrm{T}\mathbf{x}_j\|_2 = \|U_{v_i}^\mathrm{T}\mathbf{x}_i - U_{v_j}^\mathrm{T}\mathbf{x}_i\|_2$$
$$\leq \|\mathbf{x}_i\|_2 \cdot \|U_{v_i} - U_{v_j}\|_F. \tag{8}$$

The cross-view consistency regularization controls $\|U_{v_i} - U_{v_j}\|_F$ which is a scaled upper bound of the coincidence discrepancy. Thus, it makes the learned asymmetric metric more mathematically principled and rigorous.

3.2 Deep Unsupervised Asymmetric Metric Embedding

Based on CAMEL and the foregoing analysis, we can further propose our framework, named the *DE*ep Clustering-based *A*symmetric *ME*tric *L*earning (*DECAMEL*). DECAMEL embeds the unsupervised asymmetric metric into the network by jointly learning the feature representation and the asymmetric metric.

In the last subsection we assume that we have extracted features for the training images. Here, we specify the feature extractor function f as a deep convolutional network, which is parameterized by Θ. So, the feature representation is given by $\mathbf{x}_i = f(\mathbf{M}_i; \Theta)$, where $\mathbf{M}$ is an image from our training image set $\mathcal{M} = \{\mathbf{M}_i, v_i\}_{i=1}^N$. Our goal is to develop an end-to-end framework, $g(\mathbf{M}_i, v_i; \Theta, \{U_v\}_{v=1}^V)$, which includes the feature extraction and the unsupervised asymmetric metric learning.

Recall the objective in Eq. (7), where in CAMEL we aim to minimize it under constraints. To avoid trivial implementation and better adapt to the back-propagating algorithm in the optimization procedure, we develop a soft version of constraints, by replacing the constraints with a regularization term $f_{constraint}$ similar to $f_{consistency}$:

$$f_{constraint} = \sum_{v=1}^V \|U_v^\mathrm{T}\Sigma_v U_v - I\|_F^2. \tag{9}$$

This technology is widely used in the machine learning problems. For example, the well-known Tikhonov regularization is a soft version of the Ivanov regularization [73]. By this way, we have our loss function of DECAMEL as follow:

$$f_{loss} = f_{intra} + \lambda f_{consistency} + \gamma f_{constraint}$$
$$= \frac{1}{N}\sum_{k=1}^K\sum_{i\in\mathcal{C}_k}\|U_{v_i}^\mathrm{T}\mathbf{x}_i - \mathbf{c}_k\|^2 + \lambda\sum_{u,w}\|U_v - U_w\|_F^2$$
$$+ \gamma\sum_{v=1}^V\|U_v^\mathrm{T}\Sigma_v U_v - I\|_F^2, \tag{10}$$

where γ is the soft-constraint parameter. In our experiments we set $\gamma = 10$ to basically ensure the constraints. We empirically find this hyper-parameter needs no exhaustive tuning across datasets.

CAMEL is a linear metric learning model. Intrinsically, it cannot discover the underlying non-linear cross-view cluster structure of the feature representation. In addition, the feature extraction is separated and independent from metric in CAMEL. This leads to sub-optimality because the feature representation might have the capacity to be further improved

according to the metric. In contrast, DECAMEL addresses these problems naturally. It jointly learns the feature representation and the asymmetric metric with non-linearity capacity. Thus, it learns a better cross-view cluster structure.

3.3 Optimization and Algorithm

Before performing the gradient descent to train DECAMEL, we first learn an unsupervised asymmetric metric $\{U_v\}_{v=1}^V$ by CAMEL to initialize the metric layer of DECAMEL, as well as the cluster results $\{\mathbf{c}_k\}_{k=1}^K$. So we first introduce how to optimize the objective of CAMEL. We will see that the metric initialization by CAMEL contributes a lot to the whole framework in Section 5.4.4.

3.3.1 Metric Initialization by CAMEL

For convenience, let $\mathbf{y}_i = U_{v_i}^\mathrm{T}\mathbf{x}_i$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_N]$. We rewrite our objective function using trace instead of sum. The first term f_{intra} can be rewritten as [74]

$$\sum_{k=1}^K\sum_{i\in\mathcal{C}_k}\|U_{v_i}^\mathrm{T}\mathbf{x}_i - \mathbf{c}_k\|^2 = [\mathrm{Tr}(Y^\mathrm{T}Y) - \mathrm{Tr}(H^\mathrm{T}Y^\mathrm{T}YH)], \tag{11}$$

where

$$H = [\mathbf{h}_1, \ldots, \mathbf{h}_K], \quad \mathbf{h}_k^\mathrm{T}\mathbf{h}_l = \begin{cases} 0 & k \neq l \\ 1 & k = l \end{cases}, \tag{12}$$

and

$$\mathbf{h}_k = [0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0, 1, \ldots]^\mathrm{T}/\sqrt{n_k} \tag{13}$$

is an indicator vector with the ith entry corresponding to the instance $\mathbf{y}_i$, indicating that $\mathbf{y}_i$ is in the kth cluster if the corresponding entry is non-zero.

Now we construct an assignment function $\xi : \mathcal{X} \to \mathbb{R}^{Vd}$ where d denotes the feature dimension:

$$\xi(\{\mathbf{x}, v\}) = [\mathbf{0}^\mathrm{T}, \ldots, \mathbf{0}^\mathrm{T}, \mathbf{x}^\mathrm{T}, \mathbf{0}^\mathrm{T}, \ldots, \mathbf{0}^\mathrm{T}]^\mathrm{T}, \tag{14}$$

where $\mathbf{0}$ is a zero column vector which has the same size as $\mathbf{x}$, and $\mathbf{x}$ is placed in the vth "entry". Then we can construct $\widetilde{X} = [\widetilde{\mathbf{x}}_1, \ldots, \widetilde{\mathbf{x}}_N] \in \mathbb{R}^{Vd\times N}$, where $\widetilde{\mathbf{x}}_i = \xi(\{\mathbf{x}_i, v_i\})$. Besides, we construct

$$\widetilde{U} = [U_1^\mathrm{T}, \ldots, U_V^\mathrm{T}]^\mathrm{T}, \tag{15}$$

so that

$$Y = \widetilde{U}^\mathrm{T}\widetilde{X}. \tag{16}$$

Substitute Eqs. (16) into (11) and thus f_{intra} becomes

$$f_{intra} = \frac{1}{N}\mathrm{Tr}(\widetilde{X}^\mathrm{T}\widetilde{U}\widetilde{U}^\mathrm{T}\widetilde{X}) - \frac{1}{N}\mathrm{Tr}(H^\mathrm{T}\widetilde{X}^\mathrm{T}\widetilde{U}\widetilde{U}^\mathrm{T}\widetilde{X}H). \tag{17}$$

As for the second term, we can also rewrite $f_{consistency}$ as

$$f_{consistency} = \sum_{v,w}\|U_v - U_w\|_F^2 = \mathrm{Tr}(\widetilde{U}^\mathrm{T}D\widetilde{U}), \tag{18}$$

where

$$D = \begin{bmatrix} (V-1)I & -I & -I & \cdots & -I \\ -I & (V-1)I & -I & \cdots & -I \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -I & -I & -I & \cdots & (V-1)I \end{bmatrix} \in \mathbb{R}^{Vd\times Vd} \tag{19}$$

has $V \times V$ block entries Then, it is reasonable to relax the constraints

$$U_v^{\mathrm{T}} \Sigma_v U_v = I \quad (v = 1, \dots, V) \tag{20}$$

to

$$\sum_{v=1}^{V} U_v^{\mathrm{T}} \Sigma_v U_v = \widetilde{U}^{\mathrm{T}} \widetilde{\Sigma} \widetilde{U} = VI, \tag{21}$$

where $\widetilde{\Sigma} = diag(\Sigma_1, \dots, \Sigma_V)$, because what we expect is to prevent each U_v from shrinking to a zero matrix. The relaxed version of constraints is able to satisfy such a need, and it allows more elegant optimization. By now we can rewrite our optimization task as follows:

$$
\begin{aligned}
\min_{\widetilde{U}, H} f_{obj} = {}& \frac{1}{N} \mathrm{Tr}(\widetilde{X}^{\mathrm{T}} \widetilde{U} \widetilde{U}^{\mathrm{T}} \widetilde{X}) + \lambda \mathrm{Tr}(\widetilde{U}^{\mathrm{T}} D \widetilde{U}) \\
& - \frac{1}{N} \mathrm{Tr}(H^{\mathrm{T}} \widetilde{X}^{\mathrm{T}} \widetilde{U} \widetilde{U}^{\mathrm{T}} \widetilde{X} H) \\
s.t. \quad & \widetilde{U}^{\mathrm{T}} \widetilde{\Sigma} \widetilde{U} = VI.
\end{aligned}
\tag{22}
$$

We can easily find that our objective function is non-convex. Fortunately, in the form of Eq. (22), we can find that once H is fixed, Lagrange's method can be applied to our optimization task. And from Eq. (11), we can find that it is exactly the objective of k-means clustering [72] with respect to $\mathbf{y}_i$ once $\widetilde{U}$ is fixed. Thus, we can adopt an EM-like alternating algorithm to solve the optimization problem.

Fix H and Optimize $\widetilde{U}$. After fixing H and applying the method of Lagrange multiplier, our optimization task (22) is transformed into an eigen-decomposition problem as follow:

$$G\mathbf{u} = \eta \mathbf{u}, \tag{23}$$

where η is the Lagrange multiplier (and also is the eigenvalue here) and

$$G = \widetilde{\Sigma}^{-1}\left(\lambda D + \frac{1}{N} \widetilde{X} \widetilde{X}^{\mathrm{T}} - \frac{1}{N} \widetilde{X} H H^{\mathrm{T}} \widetilde{X}^{\mathrm{T}}\right). \tag{24}$$

$\widetilde{U}$ can be obtained by solving this eigen-decomposition problem.

Fix $\widetilde{U}$ and Optimize H. As for the optimization of H, we can simply fix $\widetilde{U}$ and conduct k-means clustering in the learned space. Each column of H, $\mathbf{h}_k$, is thus constructed by Eq. (13) according to the k-means clustering result.

3.3.2 Optimizing DECAMEL by Gradient Descent

After obtaining the initial $\{U_v\}_{v=1}^{V}$ and $\{\mathbf{c}_k\}_{k=1}^{K}$, we can optimize DECAMEL. We adopt the stochastic gradient descent (SGD) to optimize DECAMEL. The gradients are

$$\frac{\partial f_{loss}}{\partial U_v^{\mathrm{T}} \mathbf{x}} = 2(U_v^{\mathrm{T}} \mathbf{x} - \mathbf{c}_k) \tag{25}$$

and

$$\frac{\partial f_{loss}}{\partial U_v} = 2(\mathbf{x}\mathbf{x}^{\mathrm{T}} U_v - \mathbf{x}\mathbf{c}_k), \tag{26}$$

$$\frac{\partial f_{loss}}{\partial \widetilde{U}} = 2\lambda D \widetilde{U} + 4\gamma \widetilde{\Sigma} \widetilde{U}(\widetilde{U}^{\mathrm{T}} \widetilde{\Sigma} \widetilde{U} - VI). \tag{27}$$

We note that the gradient in Eq. (25) is with respect to each sample and the gradients in Eqs. (26) and (27) are with

respect to the asymmetric metric. Thus, we refer to Eq. (25) as the sample gradient and refer to Eqs. (26) and (27) as the metric gradient in the following Remark 2. We show the main algorithm of DECAMEL in Algorithm 3.3.2. Note that f_{obj} is guaranteed to converge as proved in [52]. It typically reaches convergence within 20 iterations.

Remark 2. (Explanation for Deep Metric Embedding). We can see from Eqs. (25), (26) and (27) that, the sample gradient flows over the whole network, while the metric gradient only flows to the metric. However, the metric is actually embedded into the sample gradient: according to the chain rule, the sample gradient for the feature extractor parameter Θ is

$$
\begin{aligned}
\frac{\partial f_{loss}}{\partial \Theta} &= \frac{\partial f_{loss}}{\partial f(\mathbf{M}; \Theta)} \frac{\partial f(\mathbf{M}; \Theta)}{\partial \Theta} = \frac{\partial f_{loss}}{\partial \mathbf{x}} \frac{\partial f(\mathbf{M}; \Theta)}{\partial \Theta} \\
&= \frac{\partial f_{loss}}{\partial U_v^{\mathrm{T}} \mathbf{x}} \frac{\partial U_v^{\mathrm{T}} \mathbf{x}}{\partial \mathbf{x}} \frac{\partial f(\mathbf{M}; \Theta)}{\partial \Theta} \\
&= 2(U_v^{\mathrm{T}} \mathbf{x} - \mathbf{c}_k) U_v^{\mathrm{T}} \cdot \frac{\partial f(\mathbf{M}; \Theta)}{\partial \Theta}.
\end{aligned}
\tag{28}
$$

Thus, the metric U_v^{T} is back-propagated to the whole network. As we will see in Section 4.2, the jointly learned feature bears resemblance to the metric. Furthermore, we will also see in Section 5.4.3 that this improves the cross-view discriminability of the feature. These observations seem like the metric is being "embedded" into the feature, and thus we refer to it as the "deep metric embedding".

Algorithm 1. DECAMEL

Input: The training images $\mathcal{M}$, the deep feature extractor $f(\cdot; \Theta)$
1 **Training**:
2 *Metric initialization*:
3 Extract feature representations using f to obtain the initial feature set $\mathcal{X}$.
4 Conduct k-means clustering in $\mathcal{X}$ to obtain $\{\mathbf{c}_k\}_{k=1}^{K}$ and to initialize H according to Eqs. (12) and (13).
5 Fix H and solve the eigen-decomposition problem described by Eqs. (23) and (24) to construct $\widetilde{U}$.
6 $t \leftarrow 1$ where t denotes each step in the following loop.
7 **while** $\{f_{obj}^t\}$ *not converged* **do**
- Alternate fixing $\widetilde{U}$ and H while optimizing the other.
- $t \leftarrow t + 1$.
8 **end**
9 Decompose $\widetilde{U}$ to obtain $\{U_v\}_{v=1}^{V}$.
10 Initialize the deep framework $g(\cdot, \cdot; \Theta, \{U_v\}_{v=1}^{V})$ using Θ and $\{U_v\}_{v=1}^{V}$.
11 *End-to-end joint learning*:
12 Update $\{\mathbf{c}_k\}_{k=1}^{K}$ from H according to Eqs. (12) and (13).
13 **while** *maximum iteration not reached* **do**
- Update Θ and $\{U_v\}_{v=1}^{V}$ by performing SGD using the gradients in
  Eqs. (25), (26) and (27).
- Update $\{\mathbf{c}_k\}_{k=1}^{K}$ while fixing Θ and $\{U_v\}_{v=1}^{V}$.
14 **end**
15 **Testing**:
16 *Given two testing images $\{\mathbf{M}_i, v_i\}$ and $\{\mathbf{M}_j, v_j\}$, the distance/ dissimilarity is computed by $\|g(\mathbf{M}_i, v_i) - g(\mathbf{M}_j, v_j)\|_2$.*

3.4 View Clustering: Generalizing to Unseen Views

In the beginning of this section we follow the conventional Re-ID setting that assumes the training and testing camera views are the same. However, some realistic large-scale applications might need to be view-extendable, i.e., new unseen camera views might be added to the surveillance network after training. In this case, the *generalizability* to new views becomes important. We propose a general method toward view-extendable scenarios for achieving better generalizability. The proposed method does not need to re-train the model when new views are added after training.

The main idea is that instead of learning feature transformations for each camera view, we can learn transformations for some *generalizable view prototypes*, which shall cover the most typical view-specific conditions. Then, if a new, unseen view is added after training, we can assign the new view to a view prototype and thus use the corresponding feature transformation. We elaborate our strategy in the following.

To better motivate our method, we start from defining pairwise dissimilarity/distance $d_V(\cdot, \cdot)$ of two camera views. As our model addresses view-specific bias in an overall *view level*, a straightforward idea is to define $d_V(\cdot, \cdot)$ as the distance between *distributions* of images from the two views. We adopt the simplified 2-Wasserstein distance [75], [76],[1] which has been shown effective and easy-to-compute in many vision tasks [76], [77], defined as:

$$d_V(View_u, View_v)^2 = \frac{1}{2}(\|\mathbf{m}_u - \mathbf{m}_v\|_2^2 + \|\boldsymbol{\sigma}_u - \boldsymbol{\sigma}_v\|_2^2), \quad (29)$$

where $\mathbf{m}_v$ is the mean vector of all training sample features (extracted by the feature network) from $View_v$, $\boldsymbol{\sigma}_v$ is the corresponding standard deviation vector, and $\mathbf{m}_u, \boldsymbol{\sigma}_u$ are similarly defined. From Eq. (29) we can define the *view representation* of $View_v$ as:

$$\mathbf{w}_v = [\mathbf{m}_v^{\mathrm{T}}, \boldsymbol{\sigma}_v^{\mathrm{T}}]^{\mathrm{T}}, \quad (30)$$

so that the L2 distance of the view representations is now equivalent to the distributional distance, i.e. $\frac{1}{2}\|\mathbf{w}_u - \mathbf{w}_v\|_2^2 = d_V(View_u, View_v)^2$.

With the view representation, we can model the generalizable view prototype as a cluster of views. View clusters can be generalizable and robust, since slight deviation of viewing condition is allowed in a view cluster, and hence each view cluster can cover and deal with the potential deviations of a new unseen view. For example, in a shopping mall, a newly added camera view facing a passageway may probably find a view cluster consisting of several other passageways views that share similar viewing conditions. One can use any clustering algorithm according to specific requirements, and in our method we adopt the simplest K-means clustering, as formulated by:

$$\min_{\{\mathbf{b}_j\}_{j=1}^J} f_{vc} = \frac{1}{V}\sum_{j=1}^J \sum_{v \in \mathcal{B}_j} \|\mathbf{w}_v - \mathbf{b}_j\|_2^2, \quad (31)$$

where V is the number of views, J is the number of view clusters, $\mathbf{b}_j$ denotes the centroid of the jth cluster and $\mathcal{B}_j = \{v|\mathbf{w}_v \in j\text{th cluster}\}$.

1. This simplified 2-Wasserstein distance makes a Gaussian assumption over the sample features, which is also empirically observed in our experiments. We refer the reader to [76] for further justification.

After obtaining J view clusters, we regard the view clusters as the *views* for training DECAMEL. After training, if a new view comes, we first assign this view to the most similar view cluster and use its feature transformation. We refer to this method as DECAMEL with **V**iew **C**lustering (*DECA-MEL$_{VC}$*). We summarize DECAMEL$_{VC}$ in Algorithm 2. By DECAMEL$_{VC}$, the learned feature transformations can naturally generalize to unseen views.

Algorithm 2. DECAMEL with View Clustering

Input: The training images $\mathcal{M}$, the deep feature extractor $f(\cdot; \Theta)$
1 **Training**:
2 Compute the view representations $\{\mathbf{w}_v\}_{v=1}^V$ by Eq. (30) .
3 Conduct k-means clustering in $\{\mathbf{w}_v\}_{v=1}^V$ to obtain the cluster separation $\mathcal{B}_j = \{v|\mathbf{w}_v \in j\text{th cluster}\}$.
4 For each image M_i in the training set, reassign a view label $v_i' \leftarrow j$ where $v_i \in \mathcal{B}_j$ to it. So we now have $1 \le v_i' \le J$ (the number of view clusters).
5 Feed $\mathcal{M}' = \{\mathbf{M}_i, v_i'\}$ to Algorithm 3.3.2 to train a deep framework $g(\cdot, \cdot; \Theta, \{U_j\}_{j=1}^J)$.
6 Use the learned feature extractor $g(\cdot; \Theta)$ to compute view prototypes/centroids $\{\mathbf{b}_j\}_{j=1}^J$.
7 **Testing for an unseen view** u:
8 Extract the view representation $\mathbf{w}_u$ using the learned feature extractor $g(\cdot; \Theta)$.
9 Assign this view to a view prototype j where $j = \arg\min_j \|\mathbf{w}_u - \mathbf{b}_j\|_2$.
10 Assign a view label j to all testing images from this view.
11 Follow the testing procedure in Algorithm 1.

Scalability. Solving Eq. (23) requires eigen decomposition whose computational complexity is $O((Vd)^3)$ where V is the number of views and d is the feature dimension which is constant. In the view-extendable setting, we can pre-define the number of generalizable view prototypes J ($J \ll V$), leading to a constant computational complexity of the eigen decomposition in Eq. (23). In the conventional Re-ID setting where training views and testing views come from the same pool, we can perform view clustering (and pre-define J) *only* for CAMEL (i.e., metric initialization). Then, the ith ($1 \le i \le J$) learned transformation is used to initialize all the view-specific transformations that belong to the views in the ith view cluster (so that in the joint learning of DECAMEL we still learn V transformations). We refer to this method as DECA-MEL *Initialized* with View Clustering (*DECAMEL$_{IVC}$*). We find that using a relatively small J for DECAMEL$_{IVC}$ could achieve very close performance as DECAMEL. The experimental result is shown in the supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2886878.

Finally we also note that DECAMEL$_{VC}$ is a generalization of both DECAMEL and its symmetric version in the conventional Re-ID setting. If we set $J = V$, the method is equivalent to original DECAMEL. On the other hand, if we set $J = 1$, the method degrade to learning a universal feature transformation. In this sense, DECAMEL$_{VC}$ allows us to flexibly control the generalizability versus ability to accurately model each specific viewing conditions, according to the given application scenario.
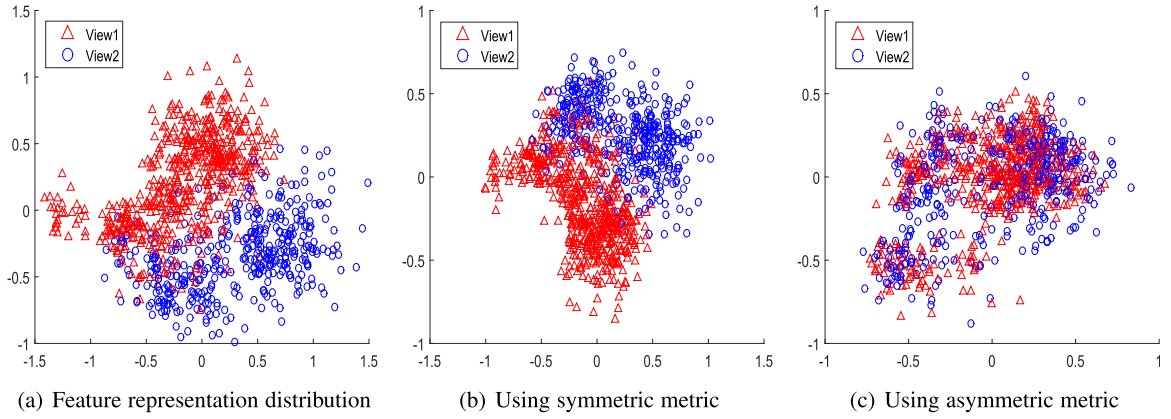
Fig. 3. Illustration of asymmetric metric alleviating view-specific bias. The data is randomly sampled from the SYSU dataset [28]. We performed PCA for visualization. Blue circles and red triangles represent data points from two camera views. *(a)* Cross-view data distribution in original feature representation space. View-specific bias is severe here, since one can easily draw a boundary to roughly separate the circles and triangles. *(b)* Distribution in the shared space after projected by the learned view-generic transformation (symmetric metric). The bias is not alleviated. *(c)* Distribution in the shared space after projected by the learned view-specific transformations (asymmetric metric). The bias is largely alleviated. (Best viewed in color).

4 INSIGHT UNDERSTANDING

In this section, we show visual results to provide intuitive perceptions on the mechanism of our framework. We first illustrate that the learned asymmetric metric can alleviate the view-specific bias in the original/initialized feature space more easily than a symmetric one. Then, we illustrate how DECAMEL progressively learns a better cross-view cluster structure based on the learned asymmetric metric by joint learning, and how it mines the potential cross-view discriminative information.

4.1 Asymmetric Metric Alleviates Views-Specific Bias

In Fig. 3, we show three cross-view data distributions, which are in the original/initialized feature space, the shared space learned by a symmetric version of CAMEL and the shared space learned by CAMEL, respectively.

We can see from Fig. 3a that in the original feature space, there is severe view-specific bias. One can easily draw a borderline to separate the data from different views. Then, in Fig. 3b, the view-specific bias is still severe in the shared space found by the learned symmetric metric (view-generic transformation), mainly because only some view-generic rotations and translations are taken for both views. In contrast, in Fig. 3c, in the shared space learned by CAMEL, the view-specific bias is alleviated so that data points from two views are much more overlapped with each other. One of the main reasons is that the learned view-specific transformations provide more flexibility to facilitate alleviating the bias.

4.2 DECAMEL Learns Cross-View Cluster Structure

In Fig. 4, we further show the cross-view distributions in both the feature space and the shared space learned by DECAMEL in different stages. By examining the distributions through stages, we can obtain an intuitive understanding for DECAMEL, in terms of how it progressively learns a better cross-view cluster structure.

Initialization. The metric initialization (i.e., CAMEL) learns a preliminary cross-view cluster structure. We first look at the leftmost column in Fig. 4. We can see that in the original feature space (the leftmost figure in the upper row), the view-specific bias is severe as we have seen in the last subsection

(data points are a subset of Fig. 3a). In contrast, after the initial metric layer (the leftmost figure in the lower row), the bias is alleviated (data points are a subset of Fig. 3c). Then, by comparing them, we can see that CAMEL learns a preliminary cross-view cluster structure, i.e., the cross-view data points representing the same identity (color) roughly get closer to each other.

Joint Learning. The joint learning facilitates learning a better cross-view cluster structure. The upper row in Fig. 4 shows the feature distributions in several stages. Through them, we can find that the feature extractor network is guided by the learned cross-view cluster structure to improve the feature representation, and in the convergence stage the view-specific bias is alleviated. This shows that the asymmetric metric is embedded into the whole network, as we discussed in Remark 2. This is mutually helpful in learning the asymmetric metric. The lower row in Fig. 4 shows the corresponding distributions after asymmetric metric layer. Through them, we can find that the asymmetric metric gradually learns a better cross-view cluster structure. Take the purple and blue identities for example. In the initial shared space (the leftmost figure in the lower row) there is around $1/3$ data points of them overlapped with each other. However, along with the joint learning procedure, they are getting more and more compact, and finally there is nearly no overlap between them in the convergence stage.

Mining Potential Discriminative Information. Finally, by comparing the initial feature distribution (the leftmost figure in the upper row) with the convergence distribution (the rightmost figure in the lower row), we can find that through the joint learning of DECAMEL, a better cross-view cluster structure is learned. Since the view-specific bias has been alleviated by the asymmetric metric, the cross-view data points belonging to the same identity can get closer to each other in the convergence stage, rather than entangling and overlapping with other identities in the initialized feature space. Therefore, during this learning procedure DECAMEL is attempting to mine the potential cross-view discriminative information. We will further show a quantitative result on evaluating the learned cross-view cluster structures in Section 5.4.1, which experimentally validates that DECAMEL learns a better cross-view cluster structure compared to using a symmetric metric.

Cross-view distribution in the **feature representation** space

Cross-view distribution in the shared space (after **asymmetric metric** layer)
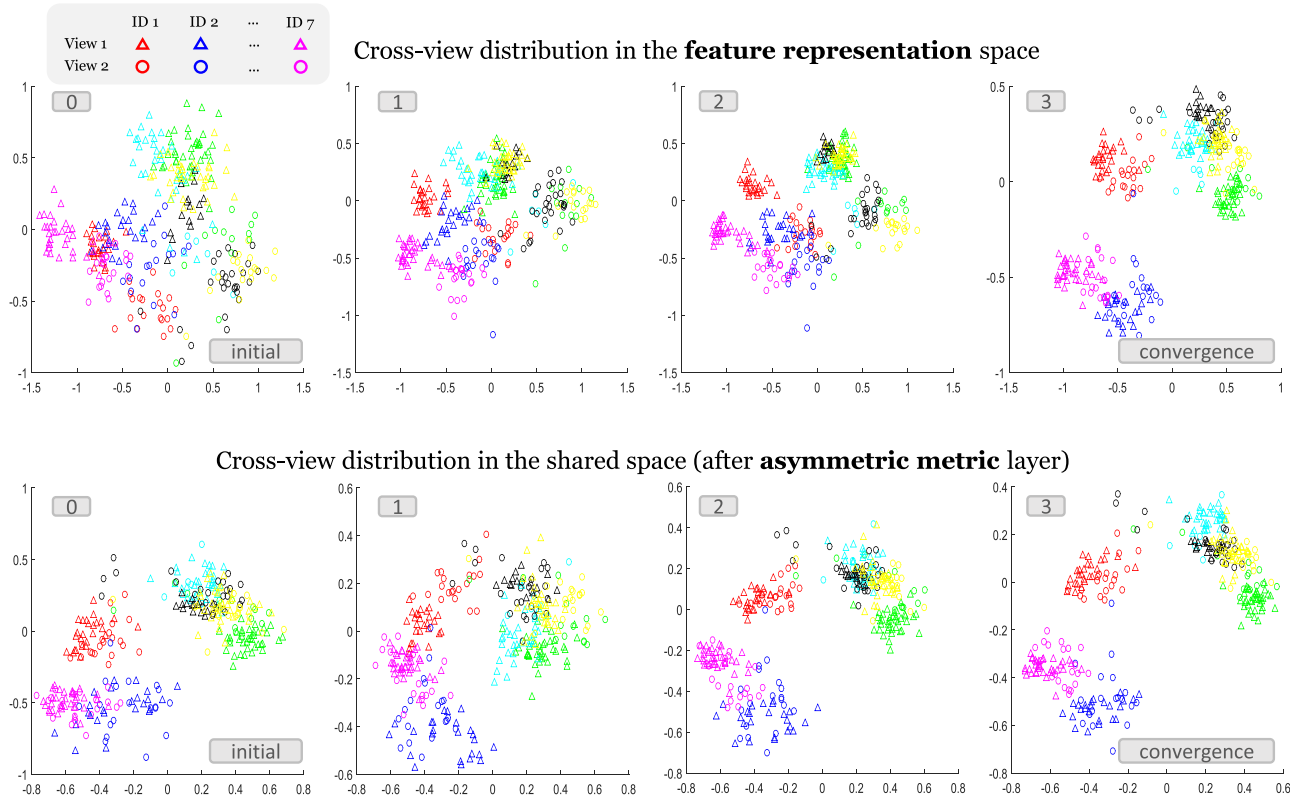

Fig. 4. Illustration of DECAMEL learning the better cross-view cluster structure via jointly learning the feature representation and the asymmetric metric. We perform PCA for visualization. Images of an identity are indicated by a specific color (e.g., all red triangles and circles are images of the first identity in the feature space). The numbers in the upper left of each figure indicates different stages, from initial to convergence. The two figures in each column are synchronous and corresponding to each other. Data points are only a subset from those in Fig. 3 for clarity. Specifically, the initial stages (the leftmost column) are subsets of Fig. 3a and 3c, respectively. (Best viewed in color and please refer to the text in Section 4.2 for more analysis. Please zoom in for better visual quality).

5 EXPERIMENTS

In this section we compare the performance of DECAMEL with other related unsupervised models to show the effectiveness. Then, we perform experimental validations and analysis to provide further comprehensive understanding of our framework.

5.1 Datasets

We conduct our comparisons on seven datasets, whose scales vary from hundreds to hundreds of thousands. In particular, since unsupervised models are more meaningful when the scale of problem is larger, our experiments are conducted on relatively bigger datasets except VIPeR [78] which is small but widely used. Various degrees of viewing condition variation can be observed in all these datasets (see Fig. 5). A brief overview of the dataset scales can be found in Table 1.

Fig. 5. Samples of the datasets. Every two images in a column are from one identity across two disjoint camera views. (a) VIPeR (b) CUHK01 (c) CUHK03 (d) SYSU (e) Market (f) ExMarket (g) MSMT17.

The VIPeR dataset [78] contains 1264 images, where every two images are captured for each identity from two camera views.

The CUHK01 dataset [79] contains 3,884 images of 971 identities captured from two disjoint views.

The CUHK03 dataset [29] contains 13,164 images of 1,360 pedestrians captured from six surveillance camera views. Pedestrian images were detected by a state-of-the-art pedestrian detector.

The SYSU dataset [28] includes 24,448 RGB images of 502 persons under two surveillance cameras. One camera view mainly captured the frontal or back views of persons, while the other observed mostly the side views.

The Market-1501 dataset [80] (Market) contains 32,668 images of 1,501 pedestrians, each of which was captured by at most six cameras. All of the images were cropped by a pedestrian detector. There are some badly-detected samples in this datasets as distractors as well.

The ExMarket dataset. Unsupervised models are more meaningful when the problem scale is larger due to the

TABLE 1
Overview of Dataset Scales

Dataset	VIPeR	CUHK01	CUHK03	SYSU	Market	ExMarket	MSMT17
# Samples	1,264	3,884	13,164	24,448	32,668	236,696	126,441
# Views	2	2	6	2	6	6	15

"#" means "the number of".

difficulty in labelling substantial cross-view data. In order to evaluate unsupervised Re-ID methods on an even larger scale, we further combined *the MARS dataset* [81] with Market. MARS is a video-based Re-ID dataset which contains 20,715 tracklets of 1,261 pedestrians. All the identities from MARS are of a subset of those from Market. We then took 20 percent frames (each one in every five successive frames) from the tracklets and combined them with Market to obtain an extended version of Market (*ExMarket*). The imbalance between the numbers of samples from the 1,261 persons and other 240 persons makes this dataset more challenging and realistic. There are 236,696 images in ExMarket in total, and 112,351 images of them are of training set.

The MSMT17 Dataset [82] is currently the most large-scale dataset which contains 126,441 images of 4,101 persons captured from 15 camera views during four days. Extreme lighting variations can be observed across camera views.

5.2 Settings

Experimental Protocols. We follow a widely adopted protocol on VIPeR [19], i.e., randomly dividing the images into two halves, one of which is used as training set and the other as test set. This procedure is repeated 10 times to offer an average performance. This dataset only allows single-shot experiments.

The experimental protocol for CUHK01 was the same as that in [19]. We randomly selected images from 485 persons to form the training set and images from the rest 486 persons formed the testing set. The evaluating procedure is repeated 10 times. We perform both multi-shot and single-shot experiments. That is, in the single-shot setting only one image of each gallery person is used for evaluation, whereas in the multi-shot setting all the images of each gallery person are used. In both settings, all probe images are used.

The CUHK03 dataset is provided together with its recommended evaluating protocol [29]. We follow the provided protocol, where images of 1,260 persons are chosen as the training set, and the remainders as testing set. This procedure is repeated 10 times. Both multi-shot and single-shot experiments are conducted.

As for the SYSU dataset, we randomly pick half pedestrians as training set and the others as testing set. In the testing stage, we basically follow the protocol as in [28]. That is, we randomly choose one and three images of each pedestrian as gallery for single-shot and multi-shot experiments, respectively. We repeat the testing procedure by 10 times.

Market is somewhat different from others. The evaluation protocol is also provided along with the dataset [80]. Since images of one person come from at most six views, the provided protocol does not adopt the single-shot setting. Instead, the protocol adopts the multi-shot setting and requires both the cumulative matching characteristic (CMC) and the mean average precision (MAP) [80]. The protocol of ExMarket is identical to Market since the identities from both datasets are completely the same as we mentioned above. For MSMT17 [82] we also use the provided protocol.

Implementation Details. We adopt the 56-layer ResNet [84] as the feature extractor network where the dimension is 64. The network is pre-trained using the JSTL pre-training technique proposed in [32], which used softmax loss with a concatenation of several Re-ID datasets including CUHK03 [29], CUHK01 [79], PRID [85], VIPeR [78], 3DPeS [86], i-

LIDS [87] and Shinpuhkan [88]. We do not exploit fine-tuning or the domain guided dropout proposed in [32]. Note that we do not use any label of the target dataset. For example, when pre-training the feature extractor network for the CUHK03 dataset, we exclude all the training samples from the CUHK03 dataset, to guarantee an unsupervised setting [52]. After pre-training, we remove the last fully-connected layer and take the output of the second-last global average pooling layer [84] as our feature. Also note that for MSMT17 [82], since it is highly challenging for unsupervised setting due to extreme lighting variations, we add Market-1501, SYSU and the Duke dataset [89], [90] to its pre-training set to improve the baseline feature.

We set $\lambda = 0.01$, $K = 500$ and fix them for all the datasets in the following comparisons. We also show a parameter evaluation on these major hyper parameters which are corresponding to certain characteristics of DECAMEL. We typically set the SGD iterations to 10,000 and the learning rate to 0.005 which is divided by 5 after 5,000 iterations. No weight decay (L_2 regularization) is applied. The batch size is 216. To guarantee that each batch contains all views, we first compute the distribution of the numbers of samples in each view in the training set (say we have two views and the distribution is [0.4, 0.6]), and then we randomly sample 0.4*216 = 86 images in the first view and 0.6*216 = 130 images in the second view. We note that we can also use standard random sampling and this empirically does not affect the performance, as shown in the supplementary material, available online. The framework is implemented based on the MatConvNet [91].[2]

5.3 Comparison to Related Unsupervised Models

Comparison to Related Unsupervised Re-ID Models. We first compare DECAMEL with the unsupervised Re-ID models. For a more fair and comprehensive comparison, we conduct experiments on the seven datasets for DECAMEL and the code-available related models. The compared models in the following comparisons adopt the same baseline JSTL feature which is used for initialization in DECAMEL. We have tuned the hyper parameters for the compared models to adapt to the JSTL feature, and thus report even better results than the original in literatures [46], [47], [48] (the performances are worse than the original without this tuning procedure). We use the available code for the sparse dictionary learning model (denoted as DIC) [48], the sparse iterative re-ranking model ISR [46], the CCA-based kernel subspace learning model RKSL [47], and sparse auto-encoder (SAE) [83]. We also compare our model with the baseline JSTL feature [32], which adopts the euclidean distance as its metric. The comparative results are measured by the cumulative characteristic curve and the rank-1 matching rate of CMC. We show the matching rate in Table 2, and show the CMC in Fig. 6.

As reported in Table 2, our model outperforms other models on all the datasets in both settings. In addition, from Fig. 6, our model outperforms other models by large margins at any rank. This is partly because DECAMEL explicitly deals with the view-specific bias problem by learning an asymmetric metric. Note that the improvements are notably significant on CUHK01 and SYSU. We can see that the

---

2. Demo code is available in https://github.com/KovenYu/DECAMEL.

TABLE 2
Comparison with Related Unsupervised Models: Single-Shot ("Single") and Multi-Shot ("Multi") Rank-1
Matching Rate and MAP in Percentage

Dataset	VIPeR	CUHK01	CUHK01	CUHK03	CUHK03	SYSU	SYSU	Market	ExMarket	MSMT17
Measure	single	single	multi	single	multi	single	multi	multi(MAP)	multi(MAP)	multi(MAP)
DIC [48]	**29.94**	49.31	52.85	27.38	36.51	21.28	28.56	**50.21(22.68)**	**52.18(21.19)**	**22.81(7.01)**
ISR [46]	27.53	**53.17**	**55.66**	**31.13**	**38.50**	23.16	33.77	40.32(14.27)	42.99(15.74)	21.50(6.10)
RKSL [47]	25.76	45.41	50.13	25.79	34.75	17.64	23.01	33.97(11.03)	34.86(10.40)	15.41(4.30)
SAE [83]	20.70	45.33	49.94	21.18	30.51	18.02	24.15	42.40(16.23)	43.97(15.10)	19.29(5.50)
JSTL [32]	25.73	46.26	50.61	24.66	33.15	19.92	25.59	44.69(18.36)	46.41(16.68)	21.24(6.05)
AML [54]	23.10	46.78	51.14	22.19	31.41	20.88	26.39	44.71(18.36)	46.20(16.22)	21.16(6.08)
UsNCA [55]	24.27	47.01	51.70	19.76	29.59	21.07	27.18	45.22(18.91)	47.03(16.91)	22.01(6.53)
DECAMEL	**34.15**	**65.81**	**69.00**	**38.27**	**45.82**	**36.14**	**43.90**	**60.24(32.44)**	**62.98(33.28)**	**30.34(11.13)**

*In each column, the best is indicated in **red** and the second in **blue**.*

cross-view condition variations are particularly severe on these two datasets as shown in Fig. 5b and 5d intuitively. For example, the changes of illumination are extremely severe in Fig. 5b and 5d, and the differences between features from the two views may be caused more by illumination than by identity under such a situation. In particular, although the CCA-based model RKSL also produces specific feature projections for different views, it learns the specific feature projections inconsistently, and thus does not deal with the view-specific bias. Apart from this, DECAMEL (and CAMEL as reported in Table 5) learns a compact cross-view cluster structure for mining potential discriminative information while RKSL does not.

Comparison to Published State-of-the-Art Results. Now we compare our model with the reported results in published literatures, including the transfer learning model UDML [49], the hand-crafted feature model SDALF [42], the graph learning model GL [50] and the saliency learning model GTS [45] and SDC [44]. We show the comparative results in Table 3. Note that these models have not been evaluated on SYSU, Market and ExMarket, so we can only compare with their reported results on VIPeR, CUHK01 and CUHK03 (all single-shot). As shown in Table 3, DECAMEL outperforms these models.

Comparison to Clustering-Based Metric Models. We also compare with a typical clustering-based metric learning model AML [54], and a recently proposed one UsNCA [55]. As reported in Table 2 and Fig. 6, DECAMEL can also achieve notable improvements over them. A main reason should be that DECAMEL learns an asymmetric metric to address the view-specific bias problem so as to learn a better cross-view cluster structure. In contrast, the compared clustering-based models do not take into consideration this issue which is particularly important for Re-ID.

5.4 Further Analysis of DECAMEL

In the following, we provide some further experimental validations and analysis to make a more comprehensive understanding for the framework components and show some significant properties.

5.4.1 Asymmetric versus Symmetric Modelling

We first evaluate the asymmetric modelling in our framework. To this end, we develop a symmetric version of DECAMEL and denote it as DECMEL. The only difference between them is that DECMEL learns (and embeds) a symmetric metric instead of an asymmetric one. We show the comparative results of performances in the upper part of
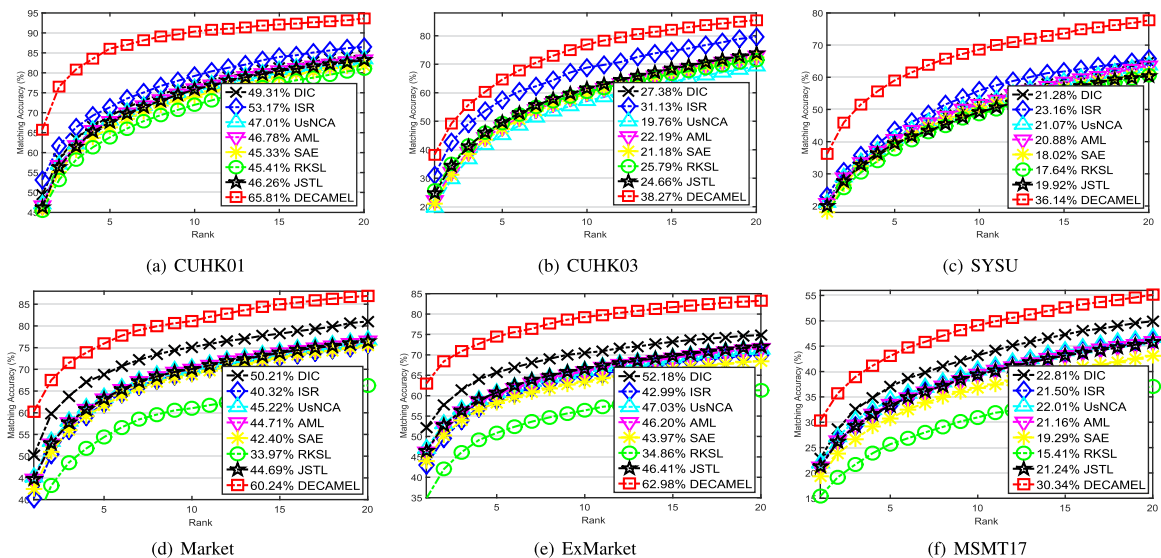

Fig. 6. CMC curves for comparisons with related unsupervised models. In each legend, the figure beside the model name is the rank-1 matching rate. For clarity, we omit VIPeR and show the single-shot results for CUHK01, CUHK03 and SYSU.

TABLE 3
Comparison with the State-of-the-Art Unsupervised Re-ID
Models Reported in Literature

Model	SDALF [42]	UDML [49]	GL [50]	SDC [44]	GTS [45]	DECAMEL
VIPeR	19.9	31.5	**33.5**	26.7	25.2	**34.2**
CUHK01	9.9	27.1	**41.0**	26.6	-	**65.8**
CUHK03	4.9	-	**30.4**	7.7	-	**38.3**

*The performance is measured by rank-1 matching rate (%) in single-shot set-ting. "-" means no reported result. In each row, the best is indicated in **red** and the second in **blue***

Table 4. We can see that DECAMEL achieves much higher performances than DECMEL.

To further explore the differences between DECAMEL and DECMEL, we also develop a natural, reasonable measure to evaluate the learned cross-view cluster structure in terms of mining the cross-view discriminative information. The measure is formulated as

$$S = inter/intra$$
$$= \left(\frac{1}{P(P-1)}\sum_{p \neq q}^{P} \|\mathbf{d}_p - \mathbf{d}_q\|_2\right) \Bigg/ \left(\frac{1}{T}\sum_{p=1}^{P}\sum_{i \in \mathcal{D}_p} \|\mathbf{x}_i - \mathbf{d}_p\|_2\right),$$
(32)

where P denotes the number of persons, $\mathcal{D}_p$ is a set containing the indexes of all the cross-view images of the pth person, $\mathbf{d}_p$ denotes the centroid of the pth person, and T denotes the total number of cross-view person images. In Eq. (32), the numerator measures the inter-person discrepancy and the denominator measures the intra-person discrepancy. We note that we only use the label information to form the measure to evaluate how discriminative the learned cross-view cluster structure is for Re-ID, i.e., the higher the S value is, the more easily we can distinguish different persons. We show the comparative results in the lower part of Table 4. We can see in Table 4 that DECAMEL has higher S values on all datasets, indicating that the asymmetric modelling in our framework helps learn a better cross-view cluster structure to facilitate mining the potential cross-view discriminative information. This can be one of the main reasons why DECAMEL outperforms DECMEL. In the following parameter evaluation we will further explore the behavior of the proposed asymmetric modelling.

TABLE 4
Evaluation of the Asymmetric Modelling in Our Framework

Dataset	CUHK01	CUHK03	SYSU	Market	ExMarket	MSMT17
Measure	single	single	single	multi(MAP)	multi(MAP)	multi(MAP)
DECMEL	55.95	27.86	25.38	49.94(23.15)	53.00(26.53)	23.88(8.01)
DECAMEL	**65.81**	**38.27**	**36.14**	**60.24(32.44)**	**62.98(33.28)**	**30.34(11.13)**
Measure	S	S	S	S	S	S
DECMEL	2.13	1.44	1.61	1.79	1.82	1.33
DECAMEL	**2.36**	**1.87**	**1.88**	**1.93**	**1.97**	**1.42**

"DECMEL" denotes the symmetric version of DECAMEL. The S value is defined in Eq. (32). The performance is measured by single-shot ("single") and multi-shot ("multi") rank-1 matching rate and MAP in percentage. For clar-ity, we drop VIPeR and the multi-shot results of CUHK01, CUHK03 and SYSU, which follow a similar pattern to single-shot results.

5.4.2 Parameter Evaluation

In the comparisons we fix the major hyper parameters, and here we discuss the behaviors of the major hyper parameters to have a better understanding of our framework. The cores of our framework are asymmetric modelling and cross-view clustering. They are characterized mainly by λ (the cross-view consistency regularizer) and K (the number of clusters), respectively.

Evaluation of λ: Characteristic of Asymmetric Modelling. The cross-view consistency regularizer, λ, controls the degree of asymmetric modelling. When λ is larger, the larger penalty enforces the discrepancy between any pair of projection basis to be smaller, and thus the asymmetric modelling will become more symmetric. We show the matching rate as a function of λ in Figs. 7a, 7b and 7c. When λ is in a median range, the performance is relatively stable. When λ is too large, the matching rate will drop. In fact, in the extreme case when λ goes to infinity, it is equivalent to the symmetric ver-sion. This shows that the asymmetric modelling is very significant in our framework.

On the other hand, when λ is too small, the matching rate also drops. To understand the reason behind, we examine the extreme case when $\lambda = 0$. This is equivalent to taking out the cross-view consistency regularization in Eqs. (7) and (10). In this extreme case, the model actually fails to learn. To reveal the underlying reason, we show the cross-view data distribu-tion in the learned shared space without the regularization in Fig. 7d. We find that the distribution collapses roughly to two lines. This is because the intrinsic consistency across the distri-butions of different views is not preserved by the arbitrarily
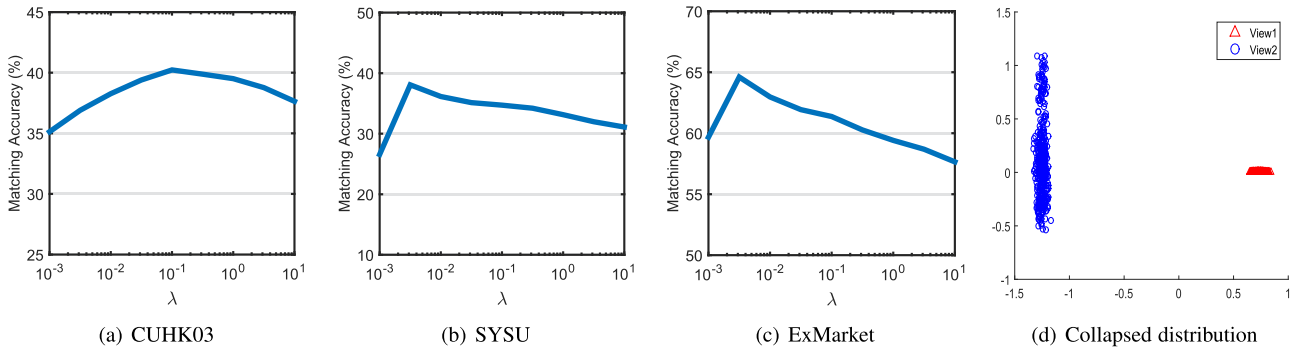

(a) CUHK03	(b) SYSU	(c) ExMarket	(d) Collapsed distribution

Fig. 7. (a)-(c) Matching rate versus λ on the three large-scale datasets. Similar observations can be made on other datasets. (d) The cross-view dis-tribution without cross-view consistency regularization. Data is from the SYSU dataset, and we performed PCA for visualization as in Fig. 3.
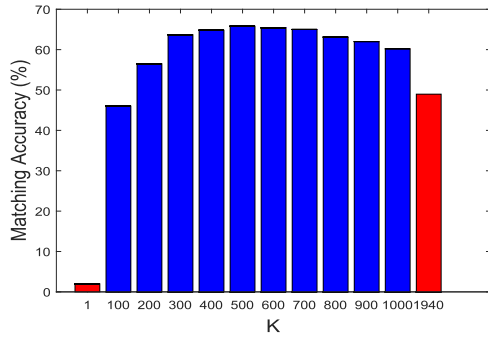
Fig. 8. Matching rate as a function of K on CUHK01. K (blue parts) is linearly spaced from 100 to 1000. We show two extreme cases (red parts) when $K = 1$ and $K = 1940$, where 1940 is the number of total training samples. Similar observations can be made on other datasets.

TABLE 5
Component-Wise Evaluation. Feat$_{init}$ Denotes the Initialized Feature

Dataset	CUHK01	CUHK03	SYSU	Market	ExMarket	MSMT17
Measure	single	single	single	multi(MAP)	multi(MAP)	multi(MAP)
Feat$_{init}$	46.26	24.66	19.92	44.69(18.36)	46.41(16.68)	21.24(6.05)
Feat$_{embd}$	**53.67**	**33.20**	**28.29**	**51.90(25.56)**	**58.37(28.21)**	**25.31(8.49)**
CAMEL	57.30	31.89	30.76	54.45(26.31)	55.88(23.88)	27.06(9.14)
DECAMEL$_f$	64.17	36.71	34.74	55.97(29.30)	60.01(31.11)	27.20(9.78)
DECAMEL$_m$	54.88	30.15	26.06	51.19(23.65)	54.81(23.09)	26.74(8.86)
DECAMEL	**65.81**	**38.27**	**36.14**	**60.24(32.44)**	**62.98(33.28)**	**30.34(11.13)**

Feat$_{embd}$ denotes the feature learned by metric embedding. DECAMEL$_f$ denotes the model when asymmetric metric layer is frozen while DECAMEL$_m$ denotes when feature extractor is frozen. Performance is measured by single-shot ("single") and multi-shot ("multi") rank-1 matching rate and MAP in percentage. For clarity, we drop VIPeR and the multi-shot results of CUHK01, CUHK03 and SYSU, which follow a similar pattern to single-shot results.

different transformations. Thus, without the cross-view consistency regularization, the learned transformations become extreme to minimize the objective and produces the collapsed cross-view distribution. Clearly, such a distribution loses the discriminative information. This observation shows that the cross-view consistency regularization averts learning a shared space with collapsed cross-view distribution.

Evaluation of K: Characteristic of Asymmetric Metric Clustering. K is the number of clusters in DECAMEL. We show the matching rate as a function of K in Fig. 8. In the middle blue parts in Fig. 8, when K is set in a median range, e.g., 300-700, the bars are tightly close to each other. This shows that to a mild extent, our framework is robust to K.

To further explore the reason behind, we show in Fig. 9 the number of clusters which contains more than one person (i.e., > 1 persons) when K varies. From Fig. 9, it is found that *(1)* despite K is varying, there is always a number of clusters containing more than one person in the initialization stage, i.e., the cluster results are in fact far from perfect. And *(2)*, in the convergence stage the numbers are consistently decreased compared to the initialization stage. This indicates that for K in a median range, the cluster results are improved consistently. This can be a reason for the mild robustness.

However, on the other hand, we can also find that in Fig. 8, the two extreme cases (red parts) where $K = 1$ and $K = 1940$ lead to performance drop. In the case when $K = 1$, the model fails to learn, somewhat similar to the situation shown in 7d, because $K = 1$ leads to a collapsed distribution where all the data points are pulled towards a single centroid.

5.4.3 Quantitative Component-Wise Evaluation

Now we discuss how framework components contribute to DECAMEL. We first set the learning rate to 0 for the metric layer to freeze it. By this way, the framework only learns the feature representation. We denote this derived model as DECAMEL$_f$. Similarly, we freeze the feature extractor, so we have DECAMEL$_m$. We also compare DECAMEL with the initialization stage (CAMEL). Besides, we compare the initialized feature representation (denoted as Feat$_{init}$) with the feature learned by metric embedding, i.e., we take the feature alone out of DECAMEL for comparison (denoted as Feat$_{embd}$). We show the results in Table 5.

From Table 5 we can make three observations: *(1)* by comparing Feat$_{init}$ with Feat$_{embd}$, we find that Feat$_{embd}$ consistently outperforms Feat$_{init}$. This shows that the deep asymmetric metric embedding also learns some underlying cross-view discriminative information for the feature representation, as have been illustrated in Section 4.2. *(2)* By comparing DECAMEL$_f$ and DECAMEL$_m$ with DECAMEL, we find that any single component cannot achieve improvements as DECAMEL. This shows that the two components of DECAMEL are intrinsically joint and cooperative, and their effects are mutually promoting each other, rather than simply linearly superposed. *(3)* By comparing CAMEL with DECAMEL, we can find that DECAMEL further provides noticeable improvements over CAMEL. This suggests the potential of jointly learning feature and metric in unsupervised Re-ID.

5.4.4 Effect of Metric Initialization

As Remark 2 reveals, the joint learning of DECAMEL is partially guided by the metric. We compare our proposed metric initialization (CAMEL) to two standard initialization strategies, i.e., identity matrix (in the view of distance metric) and randomly initialized matrix using the Xavier initialization [92] (in the view of fully-connected layer), and we denote their results as DECAMEL$_i$ and DECAMEL$_r$, respectively. As shown in Table 6, DECAMEL outperforms both of them. This is because in our learning algorithm, the metric initialization method CAMEL can learn an asymmetric distance metric which captures the cross-view person appearance variations, providing a cross-view discriminative initialization
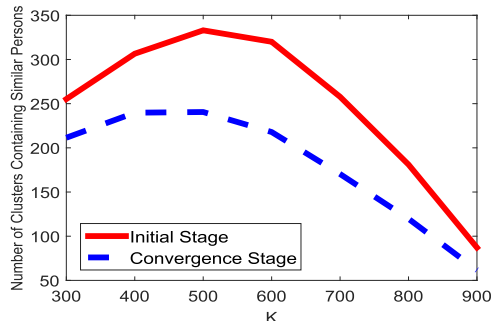

Fig. 9. Number of clusters containing more than one person at the initial stage (red solid line) and at the convergence stage (blue dashed line) when K varies on CUHK01. Similar observations can be made on other datasets.

TABLE 6
Evaluation of Different Initialization Strategies: Single-Shot
("Single") and Multi-Shot ("Multi") Rank-1 Matching Rate
and MAP in Percentage

Dataset	CUHK01	CUHK03	SYSU	Market	ExMarket	MSMT17
Measure	single	single	single	multi(MAP)	multi(MAP)	multi(MAP)
DECAMEL$_i$	48.33	28.59	18.71	40.14(15.89)	41.06(14.40)	25.65(8.32)
DECAMEL$_r$	failed	failed	failed	failed	failed	failed
DECAMEL	**65.81**	**38.27**	**36.14**	**60.24(32.44)**	**62.97(33.28)**	**30.34(11.13)**

"DECAMEL$_i$" and "DECAMEL$_r$" denote DECAMEL initialized by identity matrix and random matrix [92], respectively, rather than CAMEL. For clarity, we drop VIPeR and the multi-shot results of CUHK01, CUHK03 and SYSU, which follow a similar pattern to single-shot results.

suitable for the unsupervised feature learning, and therefore achieve superior performance.

5.4.5 Benefiting from Extra Labelled Data

We evaluate a significant property of our framework: the ability to benefit from a little extra labelled data. This is a natural way to further boost the performance for an unsupervised application scenario. We give label information to a small proportion of training samples, i.e., 10, 20 and 30 percent. These labelled samples are separated from the unlabelled samples and form extra clusters according to their labels. We show in Table 7 the results on Market which is very representative and similar observations can be made on other datasets. We can see that when given a little extra label information, the accuracy is improved as well as MAP.

5.4.6 Benefiting from More Unlabelled Data

In typical unsupervised Re-ID scenarios, e.g., public surveillance, the available data increases with time. Therefore, it is significant for an unsupervised Re-ID model to benefit from more unlabelled samples. We evaluate this property by varying the training set size on the largest dataset ExMarket, which uniquely provides over 100,000 samples. We show the results in Table 8. We can see that when the training set size grows, the accuracy and MAP are improved significantly.

5.4.7 Running Time

We report the running time of our model on the Market-1501 dataset in Table 9, compared to the most competitive models Dic [48] and ISR [46] which together take all the second places in Table 2. A GTX Titan X GPU is used for deep learning in DECAMEL. We can see that the testing procedure (i.e., usage after deployment) of DECAMEL is efficient.

TABLE 7
Evaluation When Given Label Information to a Small
Proportion of Training Samples on Market

Proportion	0%	10%	20%	30%
Acc. (MAP)	60.24(32.44)	64.55(38.26)	67.01(40.33)	69.98(43.20)

Similar observations can be made on other datasets. "Acc." is the rank-1 matching rate measured in %.

TABLE 8
Evaluation When the Training Samples Size Grows
on the Largest Dataset ExMarket

# Training samples	1,000	10,000	100,000	112,351(all)
Acc. (MAP)	52.52(22.33)	60.04(30.17)	62.86(33.21)	62.98(33.28)

"#" means the number of. "Acc." is the rank-1 matching rate measured in %.

5.5 Evaluation on the View Clustering

In the following we evaluate the View Clustering proposed in Section 3.4 for view-extendable (i.e., requiring to add new views during testing stage) and large-scale applications.

The View-Extendable Re-ID Setting. To evaluate the performance in the view-extendable scenario instead of the conventional Re-ID setting, we divide the 15 views of the MSMT17 dataset to two sets: a set of 2/3 of all the views, i.e., 10 views as training views, and a set of the other 1/3, i.e., 5 views as testing views. In the training stage, we only use the training images that are from the training views, but discard all the training images from the testing views. More specifically, we randomly divided all views into 3 subsets each of which contains 1/3 of all the views, and we alternatively select one subset as testing views and the other two as training views, repeat the above procedure for three different subset divisions (i.e., in total we train and test for 3*3 = 9 times) and report the overall averaged results. In the testing stage, we only use the probe images from the testing views for computing the quantitative results (i.e., Rank-1 accuracy and MAP). In this view-extendable setting, for comparability to the conventional setting, we remain the gallery set to contain images from all views, so that the only difference between these two settings is whether the models have seen training samples from the testing views in the training stage. We also report the results on the Market-1501 dataset on which 2 views are for testing views.

Comparative Results. Following the above view-extendable setting, we report the comparative results with the most competitive method (i.e., Dic) as well as a clustering-based model AML on the Market-1501 and MSMT17 dataset in Table 10. In our method DECAMEL with View Clustering (denoted as DECAMEL$_{VC}$ and introduced in Section 3.4) we set the number of view clusters to 10 on MSMT17 and 4 on Market-1501.

From Table 10, two observations can be made: 1) Our method also outperforms the compared methods with a clear margin in this view-extendable setting. 2) In the view-extendable setting, the performances are lower than those in the conventional setting, but this is reasonable since the training samples from the testing views are not available.

Generalizability of the View Clustering. We further investigate the generalizability of VC when more training views are

TABLE 9
Running Time on the Market-1501 Dataset

Method	Dic [48]	ISR [46]	DECAMEL
Training/Testing Time	35.2h/0.02s	-/98.0s	5.6h/0.02s

All methods are implemented in MATLAB R2017a on a Linux server. Note that ISR does not need a training procedure. Training time of DECAMEL includes all steps in Algorithm 1. Testing time is the average time of each probe image searching over the gallery list.

TABLE 10
Comparative Results in the View-Extendable Setting

Method	AML	Dic	DECAMEL$_{VC}$
Market-1501	41.57(15.01)	43.82(18.19)	56.50(29.99)
MSMT17	19.77(5.87)	21.04(6.00)	26.42(8.75)

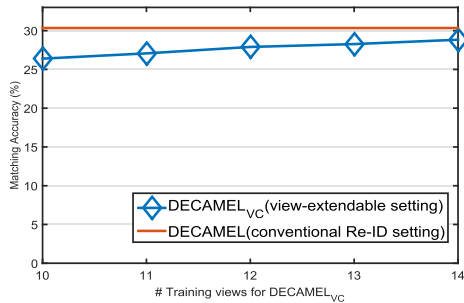*The performances of Rank-1 accuracy (MAP) are **only** of all the **unseen** views (see the text in Section 5.5).*

Fig. 10. Performances of DECAMEL$_{VC}$ in the view-extendable setting in the MSMT17 dataset. We fix $J = 10$.

available. To this end, we now take n views ($n = 5, 4, 3, 2, 1$) on the MSMT17 dataset as testing views, so that we have $(15 - n)$ training views. In the testing stage, similarly to the above setting, we report the averaged results of the unseen views. We fix the number of view prototypes $J = 10$ and show the results in Fig. 10.

From the above results, we can make two observations: 1) When there are more training views, the performance increases. This is because when there are more available training views, the view prototypes can cover a wider range of typical view-specific conditions, and thus more generalizable. 2) the performances of DECAMEL$_{VC}$ are close to DECAMEL, showing that although the testing views are not seen in the training stage, the learned projections of view prototypes can generalize to the unseen views. Note that when the number of training views is 14 DECAMEL with View Clustering (DECAMEL$_{VC}$) still has a gap of 1.5 percent compared to DECAMEL. While this observation also indicates that each view has its specific condition, DECAMEL$_{VC}$ thus strikes a balance where the ability to precisely model the view-specific condition is compromised for better generalizability. Moreover, we could infer from the above two observations that in real-world large-scale problems where there would be much more available camera views, the curve could be further extrapolated and we can expect that the performance of DECAMEL$_{VC}$ could further approximate DECAMEL, i.e., the generalizability of DECAMEL$_{VC}$ in large-scale applications shall be further improved.

6 CONCLUSION AND DISCUSSION

In this work, we present a novel approach for unsupervised Re-ID by formulating it as an unsupervised asymmetric metric learning problem. We propose a novel unsupervised loss function to produce a deep framework DECAMEL, which learns the asymmetric metric and embeds it into a deep feature learning network by end-to-end learning. The experiments show that our model can outperform the related unsupervised Re-ID models.

The analysis and experimental results suggest the effectiveness of the asymmetric modelling in unsupervised Re-ID. We note that the asymmetric modelling could be extensively embedded into other modelling strategies in unsupervised Re-ID, e.g., designing view-specific features, learning unsupervised asymmetric metrics and learning view-specific dictionaries. Our work also suggests the potential of unsupervised metric learning in Re-ID, especially that based on cross-view clustering. A future direction could be exploring the behaviour of asymmetric modelling in front of a view-imbalance problem, where the number of samples in each camera view is largely imbalanced, which could be necessary. Deriving theoretical guarantees on the robustness against the view-imbalance problem could further perfect the asymmetric modelling theoretically.

REFERENCES

[1] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, pp. 270–286, 2014.
[2] R. Vezzani, D. Baltieri, and R. Cucchiara, "People reidentification in surveillance and forensics: A survey," *ACM Comput. Surveys*, vol. 46, 2013, Art. no. 46.
[3] Y.-C. Chen, X. Zhu, W.-S. Zheng, and J.-H. Lai, "Person re-identification by camera correlation aware feature augmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 392–408, Feb. 2018.
[4] C. Liu, S. Gong, C. Loy, and X. Lin, "Person re-identification: What features are important?" in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 391–401.
[5] C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, and W. Gao, "Multi-task learning with low rank attribute embedding for person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3739–3747.
[6] Z. Shi, T. M. Hospedales, and T. Xiang, "Transferring a semantic representation for person re-identification and search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4184–4193.
[7] I. Kviatkovsky, A. Adam, and E. Rivlin, "Color invariants for person reidentification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1622–1634, Jul. 2013.
[8] R. Zhao, W. Oyang, and X. Wang, "Person re-identification by saliency learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 356–370, Feb. 2017.
[9] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang, "Similarity learning on an explicit polynomial kernel feature map for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1565–1573.
[10] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1363–1372.
[11] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, and J. Wang, "Person re-identification with correspondence structure learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3200–3208.
[12] A. Das, A. Chakraborty, and A. K. Roy-Chowdhury, "Consistent re-identification in a camera network," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 330–345.
[13] B. Ma, Y. Su, and F. Jurie, "Bicov: A novel image representation for person re-identification and face verification," in *Proc. British Mach. Vis. Conf.*, 2012, pp. 57.1–57.11.
[14] A. Mignon and F. Jurie, "Pcca: A new approach for distance learning from sparse pairwise constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2666–2672.

[15] W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by relative distance comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 653–668, Mar. 2013.

[16] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith, "Learning locally-adaptive decision functions for person verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3610–3617.

[17] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian, "Local fisher discriminant analysis for pedestrian re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3318–3325.

[18] F. Xiong, M. Gou, O. Camps, and M. Sznaier, "Person re-identification using kernel-based metric learning methods," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 1–16.

[19] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2197–2206.

[20] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2288–2295.

[21] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Learning to rank in person re-identification with metric ensembles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1846–1855.

[22] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1239–1248.

[23] D. Chen, Z. Yuan, B. Chen, and N. Zheng, "Similarity learning with spatial constraints for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1268–1277.

[24] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by discriminative selection in video ranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 12, pp. 2501–2514, Dec. 2016.

[25] H. Wang, S. Gong, X. Zhu, and T. Xiang, "Human-in-the-loop person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 405–422.

[26] L. An, M. Kafai, S. Yang, and B. Bhanu, "Person reidentification with reference descriptor," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 4, pp. 776–787, Apr. 2016.

[27] M. Hirzer, P. Roth, M. Köstinger, and H. Bischof, "Relaxed pairwise learned metric for person re-identification," *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 780–793.

[28] Y.-C. Chen, W.-S. Zheng, J.-H. Lai, and P. Yuen, "An asymmetric distance model for cross-view feature mapping in person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 8, pp. 1661–1675, Aug. 2016.

[29] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 152–159.

[30] L. Wu, C. Shen, and A. van den Hengel, "Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification," *Pattern Recognit.*, vol. 65, pp. 238–250, 2017.

[31] E. Ahmed, M. Jones, and T. K. Marks, "An improved deep learning architecture for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3908–3916.

[32] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1249–1258.

[33] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 34–39.

[34] S. Wu, Y.-C. Chen, X. Li, A.-C. Wu, J.-J. You, and W.-S. Zheng, "An enhanced deep feature representation for person re-identification," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–8.

[35] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2353–2367, May 2016.

[36] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1288–1296.

[37] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1335–1344.

[38] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 791–808.

[39] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3492–3506, Jul. 2017.

[40] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 475–491.

[41] X. Wang, W. S. Zheng, X. Li, and J. Zhang, "Cross-scenario transfer person reidentification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 8, pp. 1477–1460, Aug. 2016.

[42] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2360–2367.

[43] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino, "Custom pictorial structures for re-identification," in *Proc. British Mach. Vis. Conf.*, 2011, pp. 68.1–68.11.

[44] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3586–3593.

[45] H. Wang, S. Gong, and T. Xiang, "Unsupervised learning of generative topic saliency for person re-identification," in *Proc. British Mach. Vis. Conf.*, 2014.

[46] G. Lisanti, I. Masi, A. D. Bagdanov, and A. Del Bimbo, "Person re-identification by iterative re-weighted sparse ranking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1629–1642, Aug. 2015.

[47] H. Wang, X. Zhu, T. Xiang, and S. Gong, "Towards unsupervised open-set person re-identification," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 769–773.

[48] E. Kodirov, T. Xiang, and S. Gong, "Dictionary learning with iterative laplacian regularisation for unsupervised person re-identification," in *Proc. British Mach. Vis. Conf.*, 2015, pp. 44.1–44.12.

[49] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian, "Unsupervised cross-dataset transfer learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1306–1315.

[50] E. Kodirov, T. Xiang, Z. Fu, and S. Gong, "Person re-identification by unsupervised $l1$ graph learning," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 178–195.

[51] B. Kulis, et al., "Metric learning: A survey," in *Foundations and Trends® in Machine Learning*. Hanover, MA, USA: Now Publishers Inc., 2013.

[52] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 994–1002.

[53] H. Fan, L. Zheng, C. Yan, and Y. Yang, "Unsupervised person re-identification: Clustering and fine-tuning," *ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, vol. 14, no. 4, p. 83, 2018.

[54] J. Ye, Z. Zhao, and H. Liu, "Adaptive distance metric learning for clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–7.

[55] C. Qin, S. Song, G. Huang, and L. Zhu, "Unsupervised neighborhood component analysis for clustering," *Neurocomputing*, vol. 168, no. C, pp. 609–617, Nov. 2015.

[56] R. G. Cinbis, J. Verbeek, and C. Schmid, "Unsupervised metric learning for face identification in tv video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1559–1566.

[57] J. Jiang, B. Wang, and Z. Tu, "Unsupervised metric learning by self-smoothing operator," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 794–801.

[58] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan, "Learning consistent feature representation for cross-modal multimedia retrieval," *IEEE Trans. Multimedia*, vol. 17, no. 3, pp. 370–381, Mar. 2015.

[59] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *Int. J. Comput. Vis.*, vol. 106, no. 2, pp. 210–233, Jan. 2014.

[60] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 251–260.

[61] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, Dec. 2004.

[62] R. Rosipal and N. Krämer, "Overview and recent advances in partial least squares," in *Proc. Int. Conf. Subspace Latent Struct. Feature Select.*, 2005, pp. 34–51.

[63] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 34–39.

[64] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1875–1882.

[65] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. Int. Workshop Similarity-Based Pattern Recognit.*, 2015, pp. 84–92.

[66] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4004–4012.

[67] J. Lu, G. Wang, W. Deng, P. Moulin, and J. Zhou, "Multi-manifold deep metric learning for image set classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1137–1145.

[68] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4014–4024, Dec. 2016.

[69] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn. - Vol. 48*, 2016, pp. 478–487.

[70] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5147–5156.

[71] Z. Wang, S. Chang, J. Zhou, M. Wang, and T. S. Huang, "Learning a task-specific deep architecture for clustering," in *Proc. 2016 SIAM Int. Conf. Mining*, 2016, pp. 369–377.

[72] J. A. Hartigan, *Clustering Algorithms*. Hoboken, NJ, USA: Wiley, 1975.

[73] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998.

[74] C. H. Q. Ding and X. He, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. 2005 SIAM Int. Conf. Data Mining*, 2005, pp. 606–610.

[75] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[76] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," *arXiv preprint arXiv:1703.10717*, 2017.

[77] R. He, X. Wu, Z. Sun, and T. Tan, "Wasserstein cnn: Learning invariant features for nir-vis face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018, pp. 1–1.

[78] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *Proc. IEEE Int. Workshop Perform. Eval. Tracking Surveillance (PETS)*, vol. 3, no. 5, pp. 1–7, 2007.

[79] W. Li, R. Zhao, and X. Wang, "Human reidentification with transferred metric learning," in *Proc. 11th Asian Conf. Comput. Vis.- Vol. Part I*, 2012.

[80] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1116–1124.

[81] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian, "Mars: A video benchmark for large-scale person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 868–884.

[82] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 79–88.

[83] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area v2," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2008, pp. 873–880.

[84] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[85] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Proc. Scandinavian Conf. Image Anal.*, 2011, pp. 91–102.

[86] D. Baltieri, R. Vezzani, and R. Cucchiara, "3dpes: 3d people dataset for surveillance and forensics," in *Proc. Joint ACM Workshop Human Gesture Behavior Understanding*, 2011, pp. 59–64.

[87] W. S. Zheng, S. Gong, and T. Xiang, "Associating groups of people," in *Proc. British Mach. Vis. Conf.*, vol. 2, no. 6, 2009.

[88] Y. Kawanishi, Y. Wu, M. Mukunoki, and M. Minoh, "Shinpuhkan2014: A multi-camera pedestrian dataset for tracking people across multiple cameras," in *Proc. 20th Korea-Japan Joint Workshop Frontiers Comput. Vis.*, vol. 5, no. 7, 2014.

[89] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3774–3782.

[90] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis. Workshop Benchmarking Multi-Target Tracking*, 2016, pp. 17–35.

[91] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proc. 23rd ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[92] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

Hong-Xing Yu received the bachelor's degree in communication engineering from Sun Yat-Sen University, in 2017. He is working toward the MS degree in the School of Data and Computer Science, Sun Yat-Sen University. His research interest includes computer vision and machine learning.
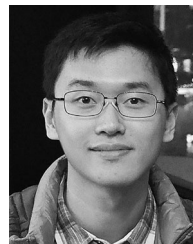
Ancong Wu received the bachelor's degree in intelligence science and technology from Sun Yat-Sen University, in 2015. He is working toward the PhD degree in the School of Electronics and Information Technology, Sun Yat-sen University. His research interests include computer vision and machine learning. He is currently focusing on the topic of person re-identification.

Wei-Shi Zheng is now a professor with Sun Yat-sen University. His research interests include person association and activity understanding in visual surveillance. He has now published more than 100 papers, including more than 70 publications in main journals (the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Neural Networks*, the *IEEE Transactions on Image Processing*, PR) and top conferences (ICCV, CVPR, IJCAI, AAAI). He served as an area chair for AVSS 2012, ICPR 2018 and BMVC 2018/2019, and a senior PC for IJCAI 2019. He has joined Microsoft Research Asia Young Faculty Visiting Programme. He is a recipient of Excellent Young Scientists Fund of the National Natural Science Foundation of China, and a recipient of Royal Society-Newton Advanced Fellowship, United Kingdom. He is an associate editor of the *Pattern Recognition Journal*.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.