Data Assimilation Network for Generalizable Person Re-Identification

Yixiu Liu, Yunzhou Zhang*, Bir Bhanu, *Life Fellow, IEEE, Sonya Coleman, Member, IEEE,Dermot Kerr*

***Abstract*—In this paper, a data assimilation network is proposed to tackle the challenges of domain generalization for person re-identification (ReID). Most of the existing research efforts only focus on single-dataset issues, and the trained models are difficult to generalize to unseen scenarios. This paper presents a distinctive idea to improve the generality of the model by assimilating three types of images: style-variant images, misaligned images and unlabeled images. The latter two are often ignored in the previous domain generalization ReID studies. In this paper, a non-local convolutional block attention module is designed for assimilating the misaligned images, and an attention adversary network is introduced to correct it. A progressive augmented memory is designed for assimilating the unlabeled images by progressive learning. Moreover, we propose an attention adversary difference loss for attention correction, and a labeling-guide discriminative embedding loss for progressive learning. Rather than designing a specific feature extractor that is robust to style shift as in most previous domain generalization work, we propose a data assimilation meta-learning procedure to train the proposed network, so that it learns to assimilate style-variant images. It is worth mentioning that we add an unlabeled augmented dataset to the source domain to tackle the domain generalization ReID tasks. Extensive experiments demonstrate that our approach significantly outperforms the state-of-the-art domain generalization methods.**

***Index Terms*—Person re-identification, data assimilation, attention correction, progressive augmented memory.**

I. Introduction

Person re-identification is a core task in video analysis and understanding, aiming to match people across disjoint camera views. Most current research focuses on solving the inherent problems in person re-identification caused by changes in background, illumination, posture, angle, *etc.* in a single-dataset. Although they have achieved excellent results, these models trained in the single-dataset tend to be inefficient on a new dataset.

There are mainly two ways to solve such a cross-dataset problem: unsupervised domain adaptation (UDA) and domain generalization (DG). UDA methods [1–8] train models in the source domain to adapt to the target domain, but they still require further updating using the unlabeled data in the target domain. Compared with UDA, DG is more challenging because the target domain is completely unseen. Our approach is dedicated to DG person ReID. There are very few prior studies [9–15] on this topic. Among them, some methods aim at learning domain-invariant feature representations, such as multi-dataset feature generalization network (MMFA-AAE) [9], multi-scale deep attention network (MuDeep) [10], instance normalization and batch normalization (DualNorm) [12], style normalization and restitution (SNR) [13], and camera-based batch normalization (CBN) [14]. Some methods focus on learning a universal mapping between a person image and its identity classifier, such as domain-invariant mapping network (DIMN) [11], or a universal matching between a person image and its deep feature maps, such as query-adaptive convolution (QAConv) [15]. Different from previous DG studies, this paper presents a distinctive idea to tackle this issue by data assimilation. "Data assimilation" means forcing the data with different characters to play the same role in the current model.

Fig. 1: Sample images that are challenging but valuable for DG person ReID.

In this paper, we force three kinds of images to join the training to improve the generality of the model. These three types of images are style-variant images, misaligned images and unlabeled images, as shown in Fig. 1. Style-variant images refer to the images with different illumination (light/dark), quality (high/low), and tone (cold/warm). The images between different datasets change significantly in style. Even in the same dataset, the style often changes due to different shooting conditions. In existing person ReID datasets, most bounding boxes of pedestrian images are manually calibrated, which consumes a lot of human resources. However, when

 It is supported by National Natural Science Foundation of China (No. 61973066), Distinguished Creative Talent Program of Liaoning Colleges and Universities (LR2019027), and Fundamental Research Funds for the Central Universities (N182608004, N2004022).

Y. Liu and Y. Zhang are with College of Information Science and Engineering, Northeastern University, Shenyang, China (e-mail: liuyixiuas-d130@gmail.com; zhangyunzhou@mail.neu.edu.cn).

B. Bhanu is with the Visualization and Intelligent Systems Laboratory (VIS-Lab), University of California, Riverside, USA (e-mail: bhanu@ee.ucr.edu).

S. Coleman and D. Kerr are with the Intelligent Systems Research Centre, Ulster University, Magee Campus, Londonderry BT48 7JL, U.K. (e-mail: sa.coleman@ulster.ac.uk; d.kerr@ulster.ac.uk).

using automatic detection algorithms, such as DPM [16] and Faster RCNN [17], to obtain the bounding box, they are often cropped incorrectly. For these misaligned images, the pedestrian is often not in the middle of the bounding box, and some of them are even incomplete or occluded. We can foresee that the style-variant and misaligned cases are also ubiquitous in the target domain. Besides, we think that the unlabeled images are also very valuable resources. It should be noted that the unlabeled images here do not come from the target domain, which is exactly different from the UDA approaches. Nevertheless, they may still contain some plots or clues similar to the images in the target domain. Based on the above explanation, we believe that the generality of the model can be improved by assimilating the three types of images shown in Fig. 1.

The existing DG methods mainly focus on style-variant images, while the misaligned and unlabeled images are often ignored. In this paper, we consider three types of images simultaneously during training. More importantly, as far as we know, we are the first to tackle the DG person ReID task using the unlabeled images. In this paper, MSMT17 [18] dataset serves as the unlabeled augmented dataset in the source domain.

A non-local convolutional block attention module (NL-CBAM) and an attention adversary network (AAN) are designed for assimilating the misaligned images. Different from the previous attention modules [19, 20], our NL-CBAM is only superimposed on the last convolutional block of bockbone instead of all/multiple convolutional blocks, which greatly reduces the complexity of the model. Excessive complexity will result in overfitting of the model in the source domain, and the generality in the target domain will be reduced accordingly. In addition, NL-CBAM can capture the remote dependence of each position in the feature maps, and the global alignment features are obtained by the subsequent global average pooling (GAP). The global alignment features are constantly improved as AAN corrects NL-CBAM. An attention adversary difference loss is proposed for attention correction. Most previous alignment methods [21–23] are based on body patch matching. They often perform poorly for the incomplete and occlusion cases shown in Fig. 1, because some body patches are missing. In contrast, the technique that improves global alignment features by attention correction is more universal, and it is more conducive to tackle the diverse misalignment scenarios in the target domain.

A progressive augmentation memory is designed for assimilating the unlabeled images. It encapsulates pseudo label estimation internally. It is very suitable for real-world DG scenario due to its scalability. Unlike the pseudo label estimation methods [2–4, 8] in UDA, once new unlabeled samples are added to the training, all the pseudo labels will be re-estimated. The progressive augmentation memory retains the last state of estimation, so that the pseudo labels of new unlabeled samples can be estimated based on the current state. The reliable pseudo samples are fed back as labeled samples, so as to augment the style-variant and misaligned samples to achieve progressive learning. A labeling-guide discriminative embedding loss is proposed for progressive learning. With the increase of assimilated unlabeled images, the generality of the model will be improved accordingly.

Instead of designing a specific feature extractor that is robust to style shift as in most previous DG methods [9, 10, 12–14], we use the existing ResNet50 as the backbone for appearance modeling, and propose a data assimilation meta-learning (DAML) procedure to train the proposed network, making it learn to assimilate style-variant images. The proposed DAML procedure is inspired by meta-learning domain generalization (MLDG) [24]. MLDG proposes to split each mini-batch into a meta-train set and a meta-test set, and simulates train/test domain shift on them to improve the generality. Although it is a homogeneous DG method, we can still use labeled samples and reliable pseudo samples with the same label to simulate domain shift. In this paper, we use the same batch split as MLDG. Different from MLDG, considering the interaction among the components, the back propagation of the network adopts a piecewise optimization strategy.

Our contributions can be summarized as the following:

- We propose a data assimilation network to tackle the DG person ReID task by assimilating three types of images. In particular, we add an unlabeled augmented dataset to the source domain to tackle this task.
- We design the NL-CBAM for assimilating the misaligned images, and introduce the AAN to correct it. The global alignment features are constantly improved as AAN corrects NL-CBAM. An attention adversary difference loss is proposed for attention correction.
- We design a scalable progressive augmentation memory, which enables the network to assimilate the unlabeled images by progressive learning. A labeling-guide discriminative embedding loss is proposed for progressive learning.
- We form a DAML procedure to train the network, making it learn to assimilate the style-variant images. In order to optimize each component of the network more reasonably, back propagation is implemented with a piecewise optimization strategy.

II. Related work

A. Single-Dataset Person ReID

Recent single-dataset person ReID approaches are dominated by the fully supervised models [20–23, 25–31]. They are trained and tested on the same dataset. These methods are dedicated to eliminating the adverse effects of view [25, 30], misalignment [20–23, 31], deformation [26–29], *etc.* on person ReID in a single-dataset. Some of them can even eliminate multiple adverse effects. For example, Liu et al. [32] propose a multi-scale triplet network to tackle the image variations such as low resolution, pose changes, occlusion, and so on. It is gratifying that they have achieved outstanding results. However, the gap between different datasets is large, especially in image style, which can be seen from the top of Fig. 1. Therefore, these state-of-the-art models often generalize poorly when applied directly to a new dataset without fine-tuning. This had led to the research directions of cross-dataset UDA and DG person ReID.

TABLE I
SUMMARY OF THE DG METHODS.

Methods	Categories	Comments*
MMFA-AAE [9]	Multi-dataset	Introduces a MMD measure to align the distributions across multiple domains
DIMN [11]	Multi-dataset	Designs a DIMN to produce a classifier using a single shot
DualNorm [12]	Multi-dataset	Jointly normalizes style and content statistics by IN and BN
SNR [13]	Single-dataset	Filters out style variations by IN, and then restitutes the removed information
CBN [14]	Single-dataset	Forces the data to fall onto the same subspace with the camera-based BN
MuDeep [10]	Single-dataset	Learns discriminative appearance features at multiple spatial scales and locations
QAConv [15]	Single-dataset	Formulates person image matching directly in deep feature maps
This paper	Both of them	Proposes a data assimilation network to assimilate three types of images to improve generality

***Abbreviations:** MMD: maximum mean discrepancy, IN: instance normalization, BN: batch normalization.

B. Cross-Dataset UDA Person ReID

UDA approaches assume that massive unlabeled data can be obtained from the target domain. The information extracted from these unlabeled data can help the model trained in the source domain adapt to the target domain. Among them, both [7] and [33] use generative adversarial networks (GANs) to generate additional training images for style transfer between different domains. Unlike [7], [33] decomposes the complicated cross-domain transfer into sub-transfers by factors, including illumination, resolution and camera view etc. [1, 5, 6] are committed to learning domain invariant features. [1] decomposes feature invariance into exemplar-invariance, camera-invariance and neighborhood-invariance, [5] unifies a local one-hot classification and a global multi-class classification into a deep network, and [6] proposes a Dissimilarity-based Maximum Mean Discrepancy (D-MMD) loss to learn domain invariant features. Moreover, some methods [2–4, 8] focus on pseudo label estimation. They assign pseudo labels to unlabeled samples through different clustering approachs, which increases the diversity of labeled samples in the source domain. For example, [4] proposes an augmented discriminative clustering (AD-Cluster) method to increase the diversity of sample clusters, which greatly improves the discrimination capability of ReID model. Compared with DG methods, UDA approaches are relatively poor in practicability, because they still need to collect unlabeled images from the target domain.

C. Cross-Dataset DG Person ReID

Previous DG person ReID methods mainly fall into two categories according to the number of datasets in the source domain. One is the multi-dataset DG methods [9, 11, 12], and the other is the single-dataset DG methods [10, 13–15]. The former trains the model on 5 large-scale datasets (CUHK02 [34], CUHK03 [35], CUHK-SYSU PersonSearch [36], Market1501 [37] and DukeMTMC-reID [38]), and tests on 4 small benchmarks (VIPeR [39], PRID [40], GRID [41] and i-LIDS [42]). The latter is mainly the transfer between 3 large-scale datasets (Market1501 [37], DukeMTMC-reID [38] and MSMT17 [18]). The principles of these methods are summarized in Table I. In this paper, we considers both multi-dataset and single-dataset cases. Our solution is quite different from the existing DG methods. We focus on three types of images that are challenging but valuable for DG person ReID, and improve the generality by assimilating them. Besides, we add an unlabeled augmented dataset (MSMT17) to the source domain to prove the feasibility of assimilating unlabeled images to improve generality.

III. DATA ASSIMILATION NETWORK

A. Overview

The training model of the proposed network in the source domain is shown in Fig. 2, while there are no AAN and progressive augmentation memory when testing in the target domain. The data flow in the network is as follows: three types of images are simultaneously loaded into the network and pass through backbone and NL-CBAM in sequence. Then, the style-variant data and misaligned data pass through AAN, and finally the possibility vectors are obtained via the classifier. The unlabeled data is stored in progressive augmentation memory. After pseudo label estimation, reliable pseudo samples can be used to augment the style-variant images and misaligned images. The training of the network follows the DAML procedure.

In Fig. 2, $\mathcal{L}_c$ and $\mathcal{R}$ are calculated for appearance modeling of labeled and unlabeled images, respectively. $\mathcal{L}_{aad}$ is calculated for attention correction, and $\mathcal{L}_{lde}$ is calculated for progressive learning. $\mathcal{L}_c$ is the cross-entropy loss, $\mathcal{L}_c = -y^T \log \hat{y}$, and $\mathcal{R}$ is the entropy-based regularization, $\mathcal{R} = \hat{y}^T \ln \hat{y}$. $\hat{y}$ denotes the output possibility vector of the classifier. For the training of labeled images, $\mathcal{L}_{lde} = 0$, and $\mathcal{R} = 0$. For the training of unlabeled images, $\mathcal{L}_c = 0$, and $\mathcal{L}_{aad} = 0$.

Traditional optimization loss is often defined as the weighted sum of all losses:

$$\mathcal{L}_{total} = \mathcal{L}_c + \lambda_1 \mathcal{L}_{aad} + \lambda_2 \mathcal{L}_{lde} + \lambda_3 \mathcal{R}, \tag{1}$$

where $\{\lambda_1, \lambda_2, \lambda_3\}$ are the weighting coefficients of losses $\{\mathcal{L}_{aad}, \mathcal{L}_{lde}, \mathcal{R}\}$. In this work, the back propagation of the model adopts a piecewise optimization strategy. We use $\mathcal{G}^b$, $\mathcal{G}^a$ and $\mathcal{G}^m$ to represent the parameters of backbone, NL-CBAM+AAN and progressive augmentation memory, respectively. $\mathcal{G}^m = \{\alpha, \alpha', \beta\}$. From the three feedforward branches shown in Fig. 2, it can be clearly seen that the backbone is shared among the three branches, while NL-CBAM+AAN and progressive augmentation memory are independent. Based on the interaction among the components, the optimization losses corresponding to $[\mathcal{G}^b, \mathcal{G}^a, \mathcal{G}^m]$ are expressed as $[\mathcal{L}_{total}, \lambda_1 \mathcal{L}_{aad}, \lambda_2 \mathcal{L}_{lde}]$.

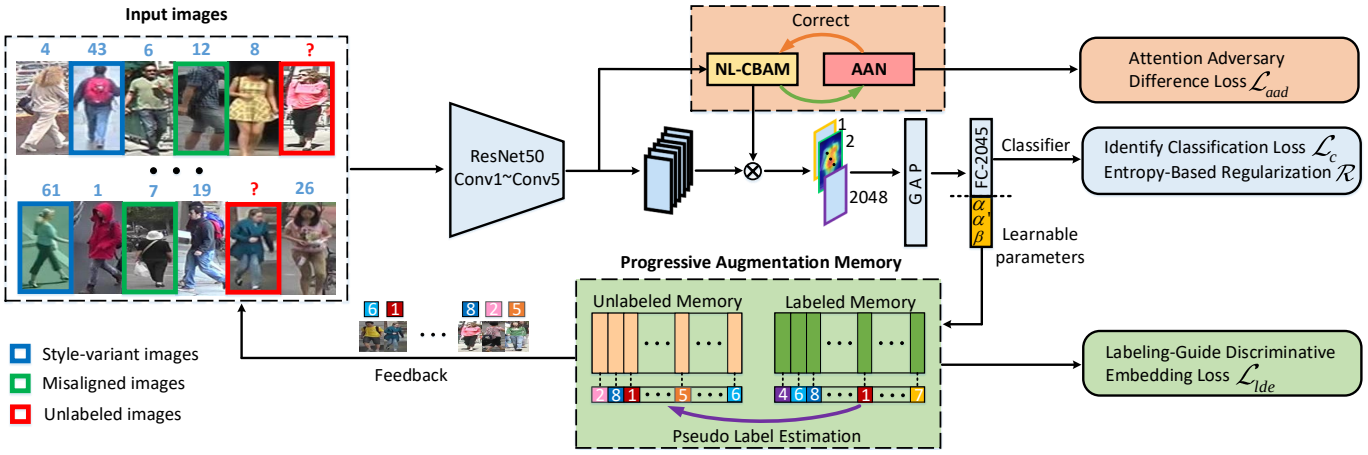
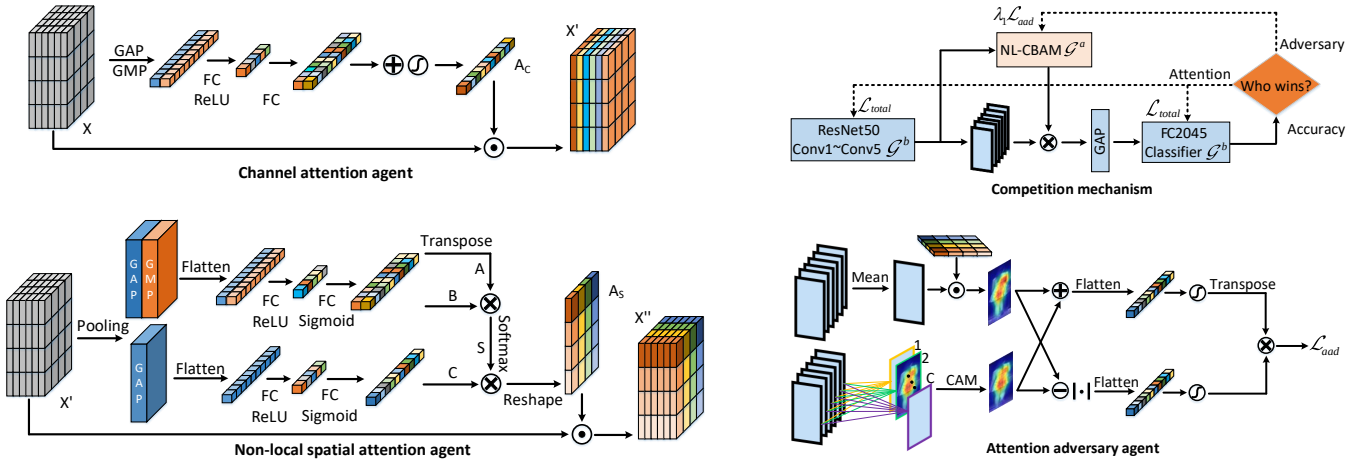


Fig. 2: Overview of data assimilation network. The input contains style-variant, misaligned and unlabeled images. Resnet50 is employed as the backbone. NL-CBAM is superimposed on its last convolution block. The global alignment features (FC-2045) are obtained by the subsequent GAP, and they are constantly improved as AAN corrects NL-CBAM. The progressive augmentation memory stores the features and (pseudo) labels of labeled images and unlabeled images, and achieves pseudo label estimation via the learnable parameters α, α', and β. The reliable pseudo samples are fed back as labeled samples to achieve progressive learning.



(a) NL-CBAM (b) AAN

Fig. 3: The architecture of NL-CBAM and AAN. $\otimes$ denotes matrix multiplication, and $|\cdot|$ denotes element-wise absolute value operation. c is the number of classes.

B. NL-CBAM and AAN

In this section, we design the NL-CBAM based on existing CBAM [19], making it more adapted to our DG task, and introduce the AAN to constantly correct NL-CBAM to further improve the global alignment features.

1) NL-CBAM: It consists of a channel attention agent and a non-local spatial attention agent. Spatial attention is concatenated behind channel attention.

Channel attention focuses on different channels of the images. It re-weights the channels of feature maps by selecting more informative ones and suppressing less useful ones. A detailed architecture of the channel attention agent is shown at the top of Fig. 3(a). The feature maps are squeezed by global average pooling (GAP) and global max pooling (GMP) on space axis, and then activated by a shared bottleneck fully-connected block. Here, we replace the multi-layer perception (MLP) [19] in CBAM with the bottleneck structure to limit model complexity. Finally, we add the activated two and get the channel attention through non-linear mapping. Given the feature maps $X \in \mathbb{R}^{(H\times W\times C)}$, where C denotes the channel and $H \times W$ is the spatial size, the channel attention A_C can be expressed as

$$A_C = \sigma(W_2^C \max(0, W_1^C X_{GAP}) + W_2^C \max(0, W_1^C X_{GMP})), \quad (2)$$

where X_{GAP} and X_{GMP} are the squeezed feature maps of X by GAP and GMP, respectively. $W_1^C \in \mathbb{R}^{\frac{C}{r}\times C}$ and $W_2^C \in \mathbb{R}^{C\times\frac{C}{r}}$ are the parameters of two FC layers, respectively. The first FC layer reduces the input dimension C by a ratio r, while the second FC layer restores the dimension. Besides, $\max(0,\cdot)$ is the ReLU activation function that exists in the first fully-connected layer. $\sigma(\cdot)$ denotes the sigmoid function. Channel attention A_C acts on the feature maps X via channel-wise multiplication $X' = X \odot A_C$.

Non-local spatial attention focuses on the position and the dependence between any two positions in the feature map. It guides the model to focus on most salient regions in the feature maps and discard irrelevant information. To capture the

remote dependence between any two positions to enhance the generality of spatial attention, we embed non-local operation [43] into spatial attention, replacing the convolution operation in original CBAM. A detailed architecture of the non-local spatial attention agent is shown at the bottom of Fig. 3(a). First, the feature maps are squeezed by two GAP and one GMP on channel axis, following by a flattening operation. One GAP branch passes through a bottleneck fully-connected block, while the other two pass through a shared bottleneck fully-connected block. The dimension reduction rate r of the above two bottleneck blocks is the same as that in channel attention. Then non-local operation is performed, and finally we obtain the non-local spatial attention. Given the feature maps X', the non-local operation and the non-local spatial attention A_S can be expressed as follows

$$\begin{aligned} A &= \sigma(W_2^S \max(0, W_1^S X_{GAP})), \\ B &= \sigma(W_2^S \max(0, W_1^S X_{GMP})), \\ C &= \sigma(W_4^S \max(0, W_3^S X_{GAP})), \\ S &= softmax(A^T B), \\ A_S &= SC, \end{aligned} \tag{3}$$

where $W_1^S, W_3^S \in \mathbb{R}^{\frac{H\times W}{r}\times(H\times W)}$, $W_2^C, W_4^S \in \mathbb{R}^{(H\times W)\times\frac{H\times W}{r}}$ are the parameters of FC layers which are similar to that in channel attention. A, B, C, S are intermediate variables. The non-local spatial attention A_S encodes the salient information of feature maps X' via element-wise production $X'' = X' \odot A_S$.

2) Attention Adversary Network: AAN is composed of a competition mechanism and an attention adversary agent. There is an attention branch and an adversary branch in the competition mechanism. The former is backbone+NL-CBAM, and the latter is backbone. They are alternately trained so that they can promote each other. Attention adversary agent is used to produce attention adversary difference loss $\mathcal{L}_{aad}$ to correct NL-CBAM.

Competition mechanism. The detailed competition mechanism is shown at the top of Fig. 3(b). The input image is classified by the model through attention branch and adversary branch. Once the branch with high accuracy wins, then the parameters of the other branch will be optimized by the corresponding loss, so that it can win in the next competition. We split the above process into the following two stages:

- Stage 1: if the adversary branch wins, freeze $\mathcal{G}^b$, optimize $\mathcal{G}^a$ using $\lambda_1\mathcal{L}_{aad}$.
- Stage 2: if the attention branch wins, freeze $\mathcal{G}^a$, optimize $\mathcal{G}^b$ using $\mathcal{L}_{total}$.

The parameter $\mathcal{G}^b$ in two branches is shared during training. Generally speaking, at the beginning of training, the adversary branch behaves better because of its lower complexity. Later, attention branch behaves better because of its stronger recognition ability for the misaligned images.

Attention adversary agent is designed to evaluate the difference between the refined feature map A_{at} and class activation map (CAM) A_{ad} of the adversary, and produce the attention adversary difference loss $\mathcal{L}_{aad}$ to correct NL-CBAM in stage 1. Because both A_{at} and A_{ad} locate the salient image regions on the feature maps, the saliency of these regions should be consistent as soon as possible in the competition. The detailed architecture of attention adversary agent is shown at the bottom of Fig. 3(b). A_{at}, A_{ad}, and $\mathcal{L}_{aad}$ can be expressed as

$$\begin{aligned} A_{at} &= \bar{X} \odot A_C \odot A_S, \\ A_{ad} &= CAM(X), \\ \mathcal{L}_{aad} &= (\sigma(A_{at} + A_{ad}))^T(\sigma|A_{at} - A_{ad}|), \end{aligned} \tag{4}$$

where $\bar{X}$ denotes the average of the feature maps on channel axis. $CAM(\cdot)$ denotes the CAM operation. It identifies the importance of the image regions by projecting the fully-connected weights between the output feature and the desired category back to the feature maps. The $1 \times 1 \times c$ convolution kernel parameters implied in CAM operation are also optimized by $\mathcal{L}_{aad}$, where c is the number of classes. The detailed derivation of CAM can be found in Reference [44]. $|A_{at} - A_{ad}|$ is the absolute deviation between A_{at} and A_{ad}, and $A_{at} + A_{ad}$ increases this deviation in salient regions of both A_{at} and A_{ad}.

C. Progressive Augmentation Memory

In this section, we will explain how progressive augmentation memory stores useful information and how to achieve progressive learning.

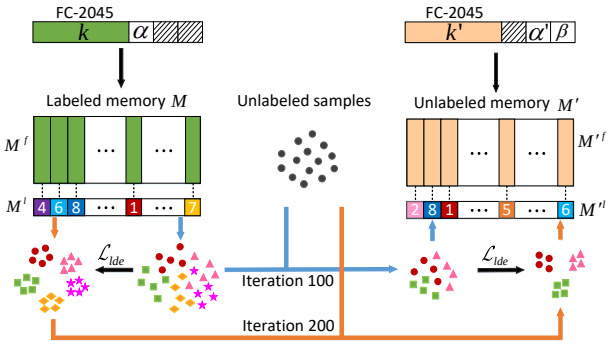


Fig. 4: The structure of memory and the pipeline of pseudo label estimation.

The structure of progressive augmentation memory and the pipeline of pseudo label estimation is shown in Fig. 4. The memory consists of a labeled memory M and an unlabeled memory M'. M is composed of a labeled feature memory M^f that stores features and a label memory M^l that stores the corresponding labels. Similarly, M' is composed of an unlabeled feature memory M'^f that stores features and a pseudo label memory M'^l that stores the corresponding pseudo labels. The number of slots of M and M' are $s1$ and $s2$, respectively.

During training, the information (features, labels) of labeled samples are written to M, and the information (features, pseudo labels) of unlabeled samples are written to M'. The pseudo labels are estimated from M, and updated as M changes. A labeling-guide discriminative embedding loss $\mathcal{L}_{lde}$ is proposed to improve pseudo label estimation. It guides M to retain the information of labeled samples useful for estimation, and M' to retain the information of reliable pseudo samples. The reliable pseudo samples are fed back as labeled samples, so as to augment the style-variant and misaligned

samples to achieve progressive learning. In this work, we utilize Least Recently Used Access (LRUA) [45] to refine $s1$ labeled samples useful for pseudo label estimation from all labeled samples, and write their information to M, refine $s2$ reliable pseudo samples from all unlabeled samples, and write their information to M'.

Writing into labeled memory. LRUA specifies that the information is written to (1) the last used slot, updating the memory with newer, possibly more relevant information, or (2) the least-used slot, preserving recently encoded information. For the t-th labeled sample, its feature k_t and label y_t are written to M_t^f and M_t^l respectively via the following weights. The usage weight W_t^u and the least-used weight W_t^{lu} are used to control the write options mentioned above. The evaluation weight W_t^e is used to evaluate the importance of each slot of M_t^f to k_t. The write weight W_t^w is used to write k_t to M_t^f. The derivation of writing is explained in detail below.

The usefulness of a slot is determined by the write and evaluation operations, and is also affected by the previous state. So W_t^u is defined as

$$W_t^u = \kappa_1 W_{t-1}^u + W_t^e + W_t^w, \tag{5}$$

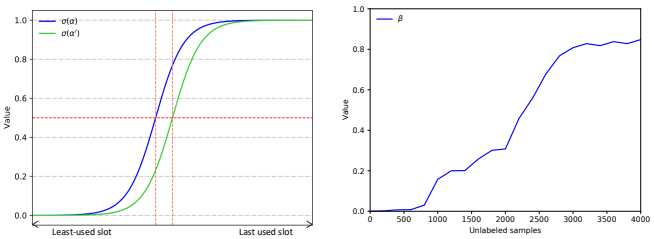
where κ_1 is the decay factor. We regard the slot that is least important to k_t as the least-used slot, so W_t^{lu} is defined as

$$W_t^{lu}(i) = \begin{cases} 1 & if\ W_t^u(i) = \min(W_t^u) \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

W_t^e evaluates the importance of each slot by calculating the distance between k_t and each unit of M_t^f, and then find the index of M_t^l where y_t is written. W_t^e and the write index I are expressed as

$$\begin{aligned} W_t^e(i) &= \frac{\exp(\cos(k_t, M_t^f(i)))}{\sum_{i=1}^{s1} \exp(\cos(k_t, M_t^f(i)))}, \\ I &= \text{find}(\max(W_t^e), M_t^l), \end{aligned} \tag{7}$$

where $\cos(\cdot)$ denotes the cosine distance. $\max(\cdot)$ returns the maximum value of a vector. $\text{find}(a, b)$ returns the index of b that meets the condition a. W_t^w should consider not only the importance of each slot of M_t^f, but also the write options. W_t^w is defined as



(a) $\sigma(\alpha)$ and $\sigma(\alpha')$ (b) β

Fig. 5: The schematic curve of $\sigma(\alpha)$, $\sigma(\alpha')$, and the variation curve of β training in Market1501 dataset.

$$W_t^w = \sigma(\alpha) W_{t-1}^e + (1 - \sigma(\alpha)) W_{t-1}^{lu}, \tag{8}$$

where $\sigma(\alpha) = \frac{1}{1+e^{-\alpha}}$, and α is a learnable scalar to interpolate between the weights. Its effect on write options is shown in Fig. 5(a). The new content tends to be written either to the last used slot ($\sigma(\alpha) \to 1$) or the least-used slot ($\sigma(\alpha) \to 0$). Before writing into M, the least-used slots are cleared. The clearing operation is as follows

$$\begin{aligned} M_t^f(i) &= M_{t-1}^f(i) \cdot (1 - W_{t-1}^{lu}(i)),\ \forall i, \\ M_t^l(i) &= M_{t-1}^l(i) \cdot (1 - W_{t-1}^{lu}(i)),\ \ \forall i. \end{aligned} \tag{9}$$

Finally, k_t and y_t are written into M_t^f and M_t^l, respectively.

$$\begin{aligned} M_t^f(i) &= M_{t-1}^f(i) + W_t^w(i) k_t,\ \forall i, \\ M_t^l(i) &= \begin{cases} y_t, & \forall i,\ \ i = I \\ M_{t-1}^l(i), & \forall i,\ \ i \neq I \end{cases}. \end{aligned} \tag{10}$$

Writing into unlabeled memory. Before writing ${k'}_t$ to ${M'}_t^f$, we first obtain its pseudo label ${y'}_t$ from the label memory M_t^l via the retrieval weight W_t^r. We find the index I^r where the stored feature is most similar to ${k'}_t$ in M_t^f, and then assign the label with index I^r in M_t^l to ${y'}_t$. The above process can be expressed as

$$\begin{aligned} W_t^r(i) &= \frac{\exp(\cos({k'}_t, M_t^f(i)))}{\sum_{i=1}^{s1} \exp(\cos({k'}_t, M_t^f(i)))}, \\ I^r &= \text{find}(\max(W_t^r), M_t^l), \\ {y'}_t &= M_t^l(I^r). \end{aligned} \tag{11}$$

We have similar weights and write index (${W'}_t^u$, ${W'}_t^{lu}$, ${W'}_t^e$, ${W'}_t^w$ and I') to write ${k'}_t$ and ${y'}_t$ to ${M'}_t^f$ and ${M'}_t^l$, respectively. The writing process is similar to Eq. (5)$\sim$(10). It should be noted that the definition of ${W'}_t^e$ is slightly different. Here ${W'}_t^e$ is defined as

$${W'}_t^u = \kappa_2 {W'}_{t-1}^u + \beta {W'}_t^e + {W'}_t^w, \tag{12}$$

where κ_2 is the decay factor, similar to κ_1. β is a learnable confidence factor that represents the trust in the pseudo labels stored in ${M'}_t^l$. Its variation curve during training is shown in Fig. 5(b). It can be seen that with the increase of training samples, the credibility of the pseudo labels increase correspondingly, and finally approaches the real labels. The schematic curve of $\sigma(\alpha')$ is also shown in Fig. 5(a), which is similar to $\sigma(\alpha)$.

Progressive learning. After writing, the information of labeled samples useful for pseudo label estimation is retained in M, while the information of the reliable pseudo samples is retained in M'. Once new unlabeled samples are added to the training, their pseudo labels can be estimated based on the current state of M, and M' is updated accordingly. In order to improve the pseudo label estimation to further improve the progressive learning, $\mathcal{L}_{lde}$ is proposed based on the following two considerations:

(1) We should ensure that the inter-class distance of the features in M^f is large enough, so that the pseudo label estimation is meaningful. When assigning pseudo label ${y'}_t$ to ${k'}_t$, we divide the samples stored in M into positive samples and negative samples according to their similarity to ${k'}_t$. The samples labeled ${y'}_t$ in M are regarded as positive samples, and the others as negative samples. Accordingly, we construct

a positive set $\mathcal{P}$ and a negative set $\mathcal{N}$ as follows

$$\begin{aligned}\mathcal{P} &= \{i|M_t^l(i) = {y'}_t, \quad \forall_i\},\\ \mathcal{N} &= \{i|M_t^l(i) \neq {y'}_t, \quad \forall_i\}.\end{aligned} \tag{13}$$

(2) Some previously written pseudo labels are obviously wrong, which is not conducive to progressive learning. They have the same pseudo label as ${k'}_t$, but their features are not sufficiently similar to ${k'}_t$. Or they have the different pseudo label from ${k'}_t$, but their features are sufficiently similar to ${k'}_t$. Thereby, we construct a false positive set $\mathcal{F}_P$ and a false negative set $\mathcal{F}_N$ as follows

$$\begin{aligned}\mathcal{F}_P &= \{j|{M'}_t^l(j) = y',\ {W'}_t^e(j) \notin \max({W'}_t^e, n)\},\\ \mathcal{F}_N &= \{j|{M'}_t^l(j) \neq y',\ {W'}_t^e(j) \in \max({W'}_t^e, n)\},\end{aligned} \tag{14}$$

where n is the number of pseudo samples labeled y' in M'. $\max(v, n)$ returns the largest n elements in vector v. Before writing to M', the slots that store these false pseudo samples are cleared by ${M'}_t^f(j) = 0, {M'}_t^l(j) = 0, \forall j \in \mathcal{F}_P, \mathcal{F}_N$.

Then we formulate the labeling-guide discriminative embedding loss as

$$\mathcal{L}_{lde} = -\log \frac{\overline{P}}{\overline{P} + \overline{N}} - \beta \log \frac{\overline{F}_P}{\overline{F}_P + \overline{F}_N}\ , \tag{15}$$

where

$$\begin{cases}\overline{P} = \frac{1}{|\mathcal{P}|}\sum_{i\in\mathcal{P}} W_t^r(i),\\ \overline{N} = \frac{1}{|\mathcal{N}|}\sum_{i\in\mathcal{N}} W_t^r(i),\\ \overline{F}_P = \frac{1}{|\mathcal{F}_P|}\sum_{j\in\mathcal{F}_P} {W'}_t^e(j),\\ \overline{F}_N = \frac{1}{|\mathcal{F}_N|}\sum_{j\in\mathcal{F}_N} {W'}_t^e(j).\end{cases} \tag{16}$$

The first term of Eq. (15) increases the discrimination of the features in M, which is conducive to pseudo label estimation, while the second term reduces the error rate of estimation, which makes the pseudo samples retained in M' more reliable. Since the pseudo labels are estimated according to the similarity between the features of unlabeled samples and the features in M, the discrimination of the features in M' are also increased accordingly. The improvement of feature discrimination can be seen clearly from Fig. 4.

D. Data Assimilation Meta-Learning

The DAML procedure enables the network to learn to assimilate the style-variant images by simulating the style shift within each mini-batch. Here, each mini-batch can be regarded as a DG subtask. To produce more subtasks that contain different styles of images, all the datasets (including MSMT17) in the source domain are fused together, the samples are shuffled between each epoch, and the number of epochs is set relatively larger than that of conventional convolutional neural networks (CNN).

During training, the mini-batch S is equally split into a meta-train set $\overline{S}$ and a meta-test set $\widetilde{S}$, each of which has N_m samples. Accordingly, each iteration is divided into two stages: meta-train and meta-test. In this work, we follow the meta-optimization objective calculation and parameter update in MLDG [24], and implement back propagation with our proposed piecewise optimization strategy. Since MLDG is a homogeneous DG method, we put the feedback reliable pseudo samples and the labeled samples with the same class into $\widetilde{S}$ and $\overline{S}$ respectively, so as to simulate the style shift in a mini-batch. The parameters [$\mathcal{G}^b$, $\mathcal{G}^a$, $\mathcal{G}^m$] are optimized by the losses [$\mathcal{L}_{total}$, $\lambda_1\mathcal{L}_{aad}$, $\lambda_2\mathcal{L}_{lde}$] in each iteration. The procedure of DAML is shown in Algorithm 1. The meta-optimization

Algorithm 1: Data Assimilation Meta-Learning Procedure

Require: Source domain $\mathcal{D}_S$
Init: Meta-parameters $\mathcal{G}^b$, $\mathcal{G}^a$, $\mathcal{G}^m$.
Hyperparameters r, s_1, s_2, κ_1, κ_2, $\lambda_1, \lambda_2, \lambda_3$, η, δ_1, δ_2.
for ite in iterations **do**
  split: mini-batch $S \rightarrow \overline{S}$ and $\widetilde{S}$
  meta-train:
  Compute the losses $\mathcal{L}(\overline{S}; \mathcal{G}^b)$, $\mathcal{L}(\overline{S}; \mathcal{G}^a)$, $\mathcal{L}(\overline{S}; \mathcal{G}^m)$
  Update ${\mathcal{G}^b}' \leftarrow \frac{\partial}{\partial \mathcal{G}^b}\mathcal{L}_{total}$
  Update ${\mathcal{G}^a}' \leftarrow \frac{\partial}{\partial \mathcal{G}^a}\lambda_1\mathcal{L}_{aad}$
  Update ${\mathcal{G}^m}' \leftarrow \frac{\partial}{\partial \mathcal{G}^m}\lambda_2\mathcal{L}_{lde}$
  meta-test:
  Compute the losses $\mathcal{L}(\widetilde{S}; {\mathcal{G}^b}')$, $\mathcal{L}(\widetilde{S}; {\mathcal{G}^a}')$, $\mathcal{L}(\widetilde{S}; {\mathcal{G}^m}')$
  Update $\mathcal{G}^b \leftarrow \frac{\partial}{\partial {\mathcal{G}^b}'}\mathcal{L}_{total}$
  Update $\mathcal{G}^a \leftarrow \frac{\partial}{\partial {\mathcal{G}^a}'}\lambda_1\mathcal{L}_{aad}$
  Update $\mathcal{G}^m \leftarrow \frac{\partial}{\partial {\mathcal{G}^m}'}\lambda_2\mathcal{L}_{lde}$
end for

TABLE II
NOTATION DEFINITION OF META-PARAMETERS AND HYPERPARAMETERS.

$\mathcal{G}^b$	training weight of ResNet50 backbone
$\mathcal{G}^a$	training weight of NL-CBAM and AAN including $W_1^S \sim W_4^S, W_1^C, W_2^C$ in NL-CBAM, and FC parameters in AAN
$\mathcal{G}^m$	learnable scalar α, α', and confidence factor β
c, N	number of classes and images
r	dimension reduction ratio of NL-CBAM
s_1, s_2	number of slots of M and M'
κ_1, κ_2	decay factor of usage weights in M and M'
$\lambda_1, \lambda_2, \lambda_3$	coefficient of loss $\mathcal{L}_{aad}$, $\mathcal{L}_{lde}$ and regularization $\mathcal{R}$
η	balance parameter of meta-optimization objective
δ_1, δ_2	step size of meta-train and meta-test

objective in each mini-batch is formulated as

$$\begin{aligned}\arg\min_{\mathcal{G}} &= \overbrace{\frac{1}{N_m}\sum_{i=1}^{N_m}\mathcal{L}(\mathcal{G})}^{meta-train\ loss} + \eta \overbrace{\frac{1}{N_m}\sum_{i=1}^{N_m}\mathcal{L}(\mathcal{G}')}^{meta-test\ loss}\ ,\\ &= \tfrac{1}{N_m}\sum_{i=1}^{N_m}(\mathcal{L}(\mathcal{G}) + \eta\mathcal{L}(\mathcal{G}'))\end{aligned} \tag{17}$$

where $\mathcal{L}$ represents the losses that appear in Algorithm 1, such as $\mathcal{L}(\overline{S}; \mathcal{G}^b)$, while $\mathcal{G}$ represents the meta-parameters, such as $\mathcal{G}^b$. η is the balance parameter between the two stages. The parameters in meta-train stage and meta-test stage are updated with the stochastic gradient descent (SGD) optimizer

$$\begin{aligned}\mathcal{G}' &= \mathcal{G} - \delta_1\mathcal{L}'(\mathcal{G}),\\ \mathcal{G} &= \mathcal{G} - \delta_2\tfrac{\partial(\mathcal{L}(\overline{S},\mathcal{G}) + \eta\mathcal{L}(\widetilde{S},\mathcal{G}'))}{\partial\mathcal{G}},\end{aligned} \tag{18}$$

where δ_1 and δ_2 are the step size of meta-train and meta-test, respectively. The meta-parameters and hyperparameters used in Algorithm 1 are listed in Table II.

TABLE III
CHARACTERISTICS OF THE DATASETS USED IN THIS PAPER.

Datasets	Identities	Cameras	Images	Labeled method	Description
CUHK02 [34]	1816	10	7264	Hand	The image quality is relatively good.
CUHK03 [35]	1467	10	13164	Hand/DPM	Person detection quality is relatively good.
CUHK-SYSU PersonSearch [36]	8432	-	18184	Hand	It mimics the real scenario of person search.
Market1501 [37]	1501	6	32217	Hand/DPM	Bounding box quality is worse than CUHK03.
DukeMTMC-reID [38]	1812	8	36441	Hand	It is a heavily labeled (full frames) dataset.
VIPeR [39]	632	2	1264	Hand	Most challenging zero-shot dataset.
PRID [40]	934	2	24541	Hand	Some trajectories are not well-synchronized.
GRID [41]	1025	8	1275	Hand	The image quality is fairly poor.
i-LIDS [42]	300	2	42495	Hand	It has extremely heavy occlusion.
MSMT17 [18]	4101	15	126441	Faster RCNN	It contains lots of complicated scenarios.

IV. EXPERIMENTS

A. Datasets and Evaluation Protocols

Datasets. The datasets in the source domain should be large enough to contain a variety of style-variant, misaligned and unlabeled images. In this way, the trained DG model has robust generality. To make the test more convincing, the selected datasets in target domain should also have similar style-variant and misaligned scenarios. The characteristics of each dataset used in this paper are summarized in Table III, which presents information about the number of identities and images, the number of cameras, the labeled method and a brief description. Among them, MSMT17 dataset serves as the unlabeled augmented dataset in the source domain.

Evaluation protocols. For source/target split, the difference between our method and the existing DG methods [9, 11–15] is that our source domain contains an unlabeled augmented dataset MSMT17. The images of MSMT17 dataset in the source domain are unlabeled, and the others are labeled. To thoroughly measure our model and other baselines, we adopt the cumulative matching characteristic (CMC) and mean average precision (mAP) as the evaluation metrics.

B. Implementation

Model. The ImageNet pre-trained ResNet50 is selected as the backbone of our model. The input images are resized to 224×224. The dimension reduction ratio r in attention branch is set to 12. The last layer of the backbone outputs a 2048 dimension vector, of which 2045 dimension is stored as the key in the feature memory M^f or M'^f, and other 3 are learnable scalars α_1, α_2, and confidence factor β. The number of slots $s1$ and $s2$ are set according to the empirical value. In this work, 9000 and 5500 are empirical values given for parameter analysis. The decay factor of usage weights κ_1 and κ_2 are equal to 0.99. The weighting coefficients of losses $\{\mathcal{L}_{aad}, \mathcal{L}_{lde}, \mathcal{R}\}$ are set to $\{0.6, 0.5, 0.2\}$. The dropout rate and the batch size are set to 0.5 and 64, respectively, and the model is trained for 500 epochs. The proposed network is implemented using the Pytorch framework on a server with 4 GeForce RTX 3090 GPUs and 96G RAM.

Optimization. The model training follows DAML procedure. The SGD optimizer is used with a momentum of 0.9 and the weight decay is set to 0.0005. We adapt a warm-up strategy to bootstrap the network to learn smoothly. The meta-train and meta-test step sizes δ_1, δ_2 at the t epochs are computed as:

$$\delta_1, \delta_2 = \begin{cases} 10^{-4} \times (\frac{t}{4} + 1), & 0 \leq t \leq 50 \\ 10^{-3}, & 50 < t \leq 200 \\ 10^{-4}, & 200 < t \leq 350 \\ 10^{-5}, & 350 < t \leq 500 \end{cases} \tag{19}$$

The balance parameter η between meta-train and meta-test loss is set to 1.0.

C. Comparison Against State-of-the-art Methods

To fully prove the superiority of our DG model, we conduct various experiments to compare with three kinds of methods: DG, UDA and fully supervised (S) methods. It should be noted that the latter two cannot be our direct competitors, because they grasp more information of the target domain. They are used to contextualize our results and set off the superiority of our model. Since previous DG methods have never used unlabeled images to train the models, in order to make fair comparisons with them, we also train our proposed network without (w/o) MSMT17 dataset, and the progressive augmentation memory is removed accordingly.

1) Comparison With DG Methods: Firstly, we compare the proposed method with the multi-dataset DG methods, and the comparison results are shown in Table IV. Our method (w/o MSMT17) follows the source/target domain split of methods [9, 11, 12]. The datasets in the source domain are CUHK02, CUHK03, Market1501, DukeMTMC-reID and CUHK-SYSU PersonSearch. The datasets in the target domain are VIPeR, PRID, GRID and i-LIDS. Our method (with MSMT17) has one more unlabeled MSMT17 dataset in the source domain. It can be seen that our method is obviously superior to the other three studies. The proposed method (w/o MSMT17) attains a 0.3% to 2.3% increase in Rank1 accuracy, only falls behind MMFA-AAE [9] by 8.2% when tested on i-LIDS dataset. In addition, it outperforms DIMN [11] on PRID (+23.5% Rank1) and GRID (+18.4% Rank1) datasets by a large margin. This success proves that it is effective to improve generality by assimilating style-variant and misaligned images. Besides, after data augmentation with the unlabeled MSMT17 dataset,

TABLE IV
COMPARISON WITH THE MULTI-DATASET DG METHODS ON VIPER, PRID, GRID AND I-LIDS (%). R: RANK. -: NO REPORT. THE FIRST/SECOND BEST RESULTS ARE MARKED IN RED/BLUE.

Method	Venue	VIPeR				PRID				GRID				i-LIDS			
		R1	R5	R10	mAP	R1	R5	R10	mAP	R1	R5	R10	mAP	R1	R5	R10	mAP
MMFA-AAE [9]	TIP21	58.4	-	-	-	57.2	-	-	-	47.4	-	-	-	84.8	-	-	-
DIMN [11]	CVPR19	51.2	70.2	76.0	60.1	39.2	67.0	76.7	52.0	29.3	53.3	65.8	41.1	70.2	89.7	94.5	78.4
DualNorm [12]	arXiv19	53.9	-	-	-	60.4	-	-	-	41.4	-	-	-	74.8	-	-	-
Ours (w/o MSMT17)	This paper	59.1	70.7	77.8	60.4	62.7	77.1	84.6	67.9	47.7	62.0	71.4	53.5	76.6	91.2	95.4	81.7
Ours (with MSMT17)	This paper	62.4	74.3	80.7	61.6	66.4	84.9	90.3	71.1	50.8	65.4	74.0	57.7	80.9	93.2	96.8	83.1

TABLE V
COMPARISON WITH THE SINGLE-DATASET DG METHODS ON MARKET1501 (M) AND DUKEMTMC-REID (D) (%). R: RANK. -: NO REPORT. THE FIRST/SECOND/THIRD BEST RESULTS ARE MARKED IN RED/BLUE/GREEN.

Methods	Venue	Source	Target: DukeMTMC-reID		Source	Target: Market1501	
			R1	mAP		R1	mAP
MuDeep [10]	TPAMI20	M	47.6	27.7	D	-	-
SNR [13]	CVPR20	M	55.1	33.6	D	66.7	33.9
CBN [14]	ECCV20	M	58.7	38.2	D	72.7	43.0
QAConv [15]	ECCV20	M	54.4	33.6	D	62.8	31.6
Ours (w/o MSMT17)	This paper	M	59.2	39.1	D	74.3	44.6
Ours (with MSMT17)	This paper	M	64.8	43.3	D	77.0	47.2

TABLE VI
COMPARISON WITH THE UDA METHODS ON MARKET1501 (M) AND DUKEMTMC-REID (D). (%). R: RANK. U: UNLABELED DATA. THE FIRST/SECOND/THIRD BEST RESULTS ARE MARKED IN RED/BLUE/GREEN.

Method	Venue	Source	Target: D		Source	Target: M	
			R1	mAP		R1	mAP
SNR+MAR [13]	CVPR20	M+D (U)	76.3	58.1	D+M (U)	82.8	61.7
AD-Cluster [8]	CVPR20	M+D (U)	72.6	54.1	D+M (U)	86.7	68.3
DAL [7]	TCSVT20	M+D (U)	75.2	57.3	D+M (U)	86.4	68.6
DCML [4]	ECCV20	M+D (U)	79.3	63.5	D+M (U)	88.2	72.3
JVTC [5]	ECCV20	M+D (U)	75.0	56.2	D+M (U)	83.8	61.1
D-MMD [6]	ECCV20	M+D (U)	63.5	46.0	D+M (U)	70.6	48.8
ECN [1]	CVPR19	M+D (U)	63.3	40.4	D+M (U)	75.1	43.0
PAST [2]	ICCV19	M+D (U)	72.4	54.3	D+M (U)	78.4	54.6
SSG [3]	ICCV19	M+D (U)	73.0	53.4	D+M (U)	80.0	58.3
Ours	This paper	M+D (U)	80.0	63.9	D+M (U)	86.6	70.8

Rank1 accuracy of our method is improved by 3.1% to 4.3%. It proves that assimilating unlabeled images is also helpful to improve generality.

Secondly, we also compare the proposed method with the single-dataset DG methods, and the comparison results are shown in the Table V. The trained models are generalized from Market1501 to DukeMTMC-reID, or from DukeMTMC-reID to Market1501. Our method (w/o MSMT17) follows the source/target domain split of methods [13–15], while an unlabeled MSMT17 dataset is added to the source domain in the "with MSMT17" setting. It can be seen that the accuracy of our method is slightly higher than the others. The proposed method (w/o MSMT17) surpasses the third best CBN [14] 0.5% (Rank1) and 0.9% (mAP) when test on DukeMTMC-reID dataset, and 1.6% (Rank1) and 0.6% (mAP) when test on Market1501 dataset. Due to the lack of various style-variant and misaligned scenarios, the improvement of our method in the single-dataset domain generalization is obviously less than that in multi-dataset domain generalization. Besides, Rank1 accuracy of our method is improved by 2.7% to 5.6% after data augmentation. It once again proves the feasibility of assimilating unlabeled images to improve generality.

2) Comparison With UDA Methods: We follow the UDA experimental settings and compare the proposed method with the UDA methods [1–8, 13] in Market1501 and DukeMTMC-reID datasets. The comparison results are shown in Table VI. In fact, it is a bit unfair to us because our model is designed on the assumption that unlabeled data is not in the target domain. Influenced by this assumption, we deliberately reduce the complexity of the model. Nevertheless, our method still achieves the best results when tested on DukeMTMC-reID dataset, outperforming the sub-optimal DCML [4] by a small margin (+0.7% Rank1, +0.4% mAP). Moreover, our method is significantly better than the methods [1–3, 5, 6, 13], and only falls behind the best DCML [4] by 1.6% and 1.5% in Rank1 and mAP when tested on Market1501 dataset. This proves that the idea of assimilating three kinds of images is also applicable to UDA tasks.

3) Comparison With Supervised Methods: Recently, some supervised methods have achieved outstanding results on some large-scale datasets, such as Market1501 and DukeMTMC-reID, but still perform poorly on small datasets. We compare our method with these state-of-the-art supervised methods [21–23, 25–29] (labeled S in Table VII) on four small datasets: VIPeR, PRID, GRID and i-LIDS. Some of them have achieved

TABLE VII
COMPARISON WITH THE FULLY SUPERVISED (S) METHODS ON FOUR SMALL DATASETS (%). R: RANK. -: NO REPORT. THE FIRST/SECOND BEST RESULTS ARE MARKED IN RED/BLUE.

Type	Method	Venue	VIPeR				PRID				GRID				i-LIDS			
			R1	R5	R10	mAP	R1	R5	R10	mAP	R1	R5	R10	mAP	R1	R5	R10	mAP
S	SSM [27]	CVPR17	53.7	-	91.5	-	-	-	-	-	27.2	-	61.2	-	-	-	-	-
S	OneShot [29]	CVPR17	34.3	-	-	-	41.4	-	-	-	-	-	-	-	51.2	-	-	-
S	SpindleNet [23]	CVPR17	53.8	74.1	83.2	-	67.0	89.0	89.0	-	-	-	-	-	66.3	86.6	91.8	-
S	MTDnet [26]	AAAI17	47.5	73.1	82.6	-	32.0	51.0	62.0	-	-	-	-	-	58.4	80.4	87.3	-
S	JLML [28]	IJCAI17	50.2	74.2	84.3	-	-	-	-	-	37.5	61.4	69.4	-	-	-	-	-
S	GOG [22]	CVPR16	49.7	79.7	88.7	-	-	-	-	-	24.7	47.0	58.4	-	-	-	-	-
S	CMDL [25]	TPAMI16	66.4	90.3	95.9	-	-	-	-	-	30.9	56.9	67.8	-	45.1	66.7	79.2	-
S	SRR_MSTC [21]	ICCV15	55.0	83.5	91.8	-	-	-	-	-	26.6	46.3	56.2	-	-	-	-	-
DG	Ours (w/o MSMT17)	This paper	59.1	70.7	77.8	60.4	62.7	77.1	84.6	67.9	47.7	62.0	71.4	53.5	76.6	91.2	95.4	81.7
DG	Ours (with MSMT17)	This paper	62.4	74.3	80.7	61.6	66.4	84.9	90.3	71.1	50.8	65.4	74.0	57.7	80.9	93.2	96.8	83.1

very high accuracy. The Rank1 of CMDL [25] on VIPeR dataset attains 66.4%, and SpindleNet [23] on PRID dataset attains 67.0%. Although they are unfair to our more challenging DG setting, our method is comparable to them and even surpass them on GRID and i-LIDS datasets. Here we just use S methods as references to set off the performance of our proposed network.

4) Comparison of Model Complexity and Time Cost: In addition to accuracy, we also compare the complexity and time cost of the models. The comparison results are shown in Table VIII. The compared models include the backbone (ResNet50), CBN and QAConv. Among them, CBN uses the same backbone as us, while QAConv uses the more complex ResNet152. We obtain the GFLOPs (Giga Floating-point Operations Per Second) and average training time (s/epoch) when training in Market1501 dataset. It can be seen that the proposed model has the highest complexity and the longest training time. Although we design the NL-CBAM and AAN, they are only superimposed on the last convolutional layer of the network, and the complexity will not increase drastically. We think that the complexity of the model mainly lies in the operations inside the progressive augmented memory, such as calculating the similarity of features.

TABLE VIII
COMPARISON OF MODEL COMPLEXITY AND TIME COST WHEN TRAINING IN MARKET1501 DATASET.

Method	GFLOPs	s/epoch
ResNet50	5.267	32.609
CBN [14]	6.233	41.061
QAConv [15]	16.718	79.654
Ours	24.534	137.908

D. Ablation Study

1) The Effect of NL-CBAM and AAN on Generality: To verify the effectiveness of NL-CBAM and AAN, we compare the incomplete models with the full model in the generalization experiment from Market1501 to DukeMTMC-reID. The models are trained w/o MSMT17 dataset, and their mAP curves are shown in Fig. 6. It can be seen that the full model (B+NL-CBAM+AAN) achieves the best results and performs much better than B (Backbone), which explains the role of NL-CBAM+AAN in assimilating the misaligned

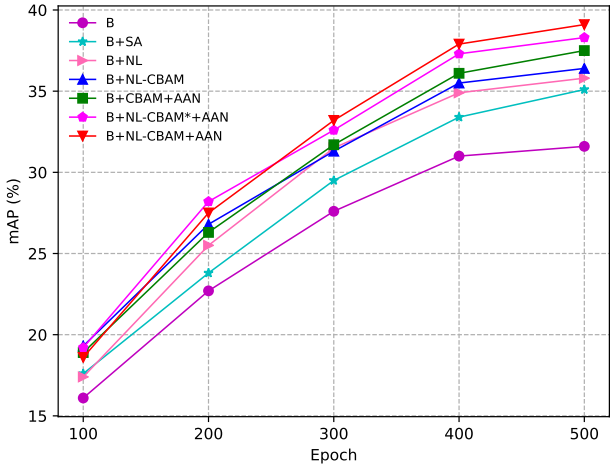
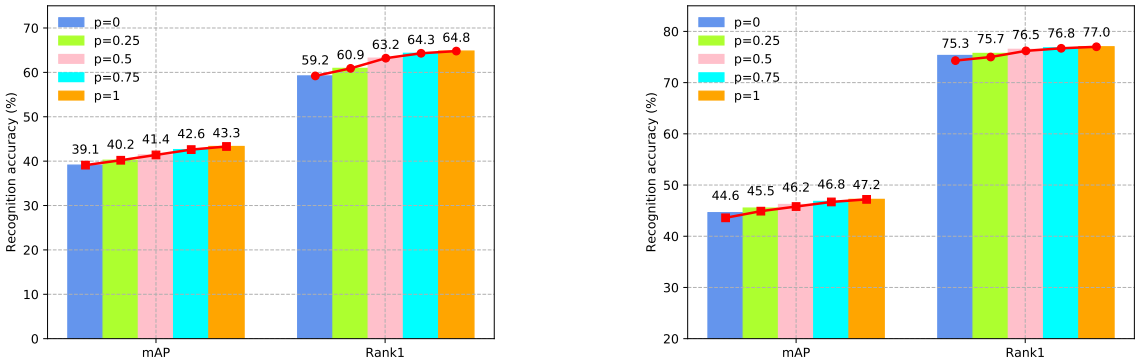


Fig. 6: Analysis of the effect of NL-CBAM and AAN components on generality.

images. Comparing the B+NL curve and the B+S curve, it can be seen that the generality of non-local operation is better than that of spatial attention (SA), which inspires us to embed non-local operation into attention module. Judging from the trend of the curves in Fig. 6, the full model is obviously better than B+CBAM+AAN, which proves that CBAM embedded with non-local operation is more suitable for DG tasks. Compared with the model without reducing the complexity (B+NL-CBAM*+AAN), the generalization performance of the full model has slight advantages, indicating the rationality of reducing the complexity of the model in DG tasks. Moreover, we can conclude that attention correction with $\mathcal{L}_{aad}$ plays an important role in domain generalization by comparing the full model curve with the B+NL-CBAM curve.

2) The Scalability and Stability of Progressive Augmentation Memory: In order to prove the stability and scalability of progressive augmentation memory, we feed the unlabeled images of MSMT17 dataset to the network in increasing proportions ($p = 0, 0.25, 0.5, 0.75$ and 1). The experimental results on Market1501 and DukeMTMC-reID datasets are shown in Fig. 7. The models trained at $p = 0$ and $p = 1$ proportions are the proposed models set at "w/o MSMT17" and "with MSMT17", respectively. As we can see from Fig. 7, with the increase of unlabeled data, the accuracy increases accordingly. The mAP accuracy of Fig. 7(a) and (b) has

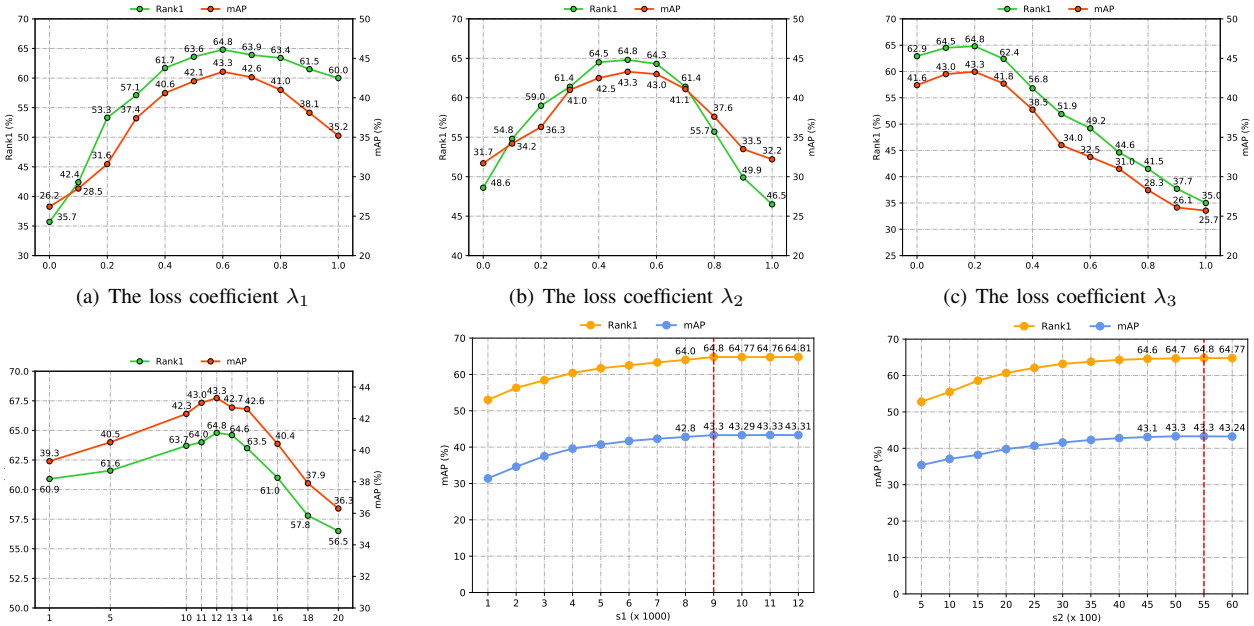


(a) Market1501→DukeMTMC-reID (b) DukeMTMC-reID→Market1501

Fig. 7: Evaluation of scalability and stability of progressive augmentation memory for generalizable person ReID.

TABLE IX
ANALYSIS OF THE EFFECT OF BATCH SPLIT AND PIECEWISE OPTIMIZATION IN DAML ON GENERALITY (%). R: RANK. BS: BATCH SPLIT. PO: PIECEWISE OPTIMIZATION.

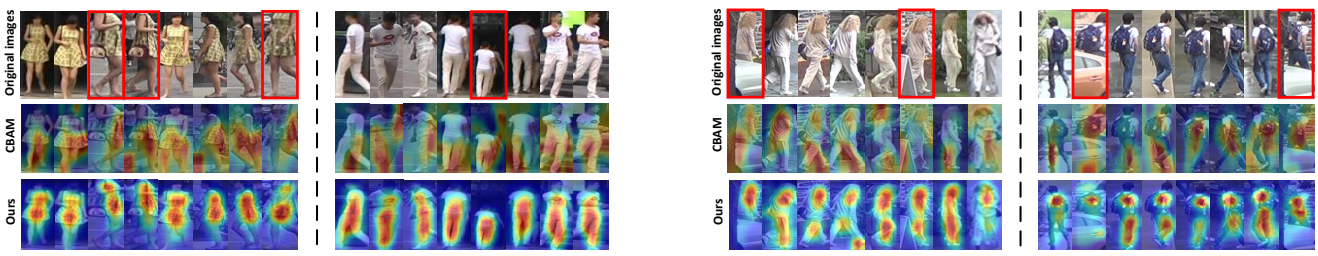
Method	BS	PO	VIPeR		PRID		GRID		i-LIDS	
			R1	mAP	R1	mAP	R1	mAP	R1	mAP
$DAML_1$			44.1	45.8	46.9	50.3	33.6	37.4	58.8	64.7
$DAML_2$	✓		59.9	58.5	64.8	69.1	47.5	53.6	76.7	78.0
$DAML_3$	✓	✓	**62.4**	**61.6**	**66.4**	**71.1**	**50.8**	**57.7**	**80.9**	**83.1**



(a) The loss coefficient λ_1 (b) The loss coefficient λ_2 (c) The loss coefficient λ_3

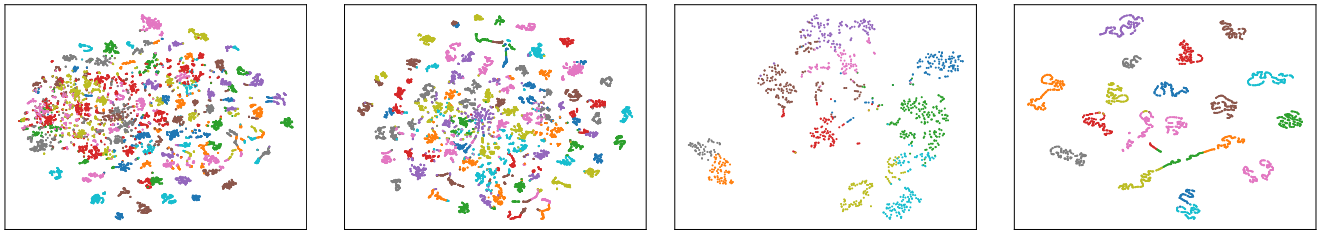
(d) The dimension reduction ratio r (e) The number of slots $s1$ (f) The number of slots $s2$

Fig. 8: Analysis of the effect of key hyperparameters on generality.



(a) Market1501 (b) DukeMTMC-reID

Fig. 9: Visualization of attention heat maps. (Top): original images. (Middle): heat maps generated by CBAM. (Bottom): heat maps generated by our NL-CBAM+AAN. The misaligned cases in Fig. 1 are framed in red.



(a) M, Iteration 500 (b) M, Iteration 1000 (c) M', Iteration 500 (d) M', Iteration 1000

Fig. 10: t-SNE visualization of the features stored in M (a, b) and M' (c, d). Points of the same color represent features from the same identity.

been improved 4.2% and 2.6%, respectively. It shows that the memory is scalable, and the generality of the model improves steadily with the increase of assimilated unlabeled images.

3) The Effect of Batch Split and Piecewise Optimization in DAML: To verify the effectiveness of batch split (BS) in MLDG [24] and the proposed piecewise optimization (PO) for our DG task, we have tried three settings of DAML to train the network (with MSMT17). The comparison results on VIPeR, PRID, GRID and i-LIDS datasets are shown in Table IX. DAML$_1$ updates the parameters with the conventional SGD and optimizes the network directly with the total loss $\mathcal{L}_{total}$. Both DAML$_2$ and DAML$_3$ split the mini-batch into a meta-train set and a meta-test set, and update the parameters with Eq. (18). DAML$_2$ optimizes the network directly with the total loss $\mathcal{L}_{total}$, while DAML$_3$ optimizes the components of the network with the piecewise loss [$\mathcal{L}_{total}$, $\lambda_1\mathcal{L}_{aad}$, $\lambda_2\mathcal{L}_{lde}$]. DAML$_3$ is the training setting of this work. As can be seen from Table IX, the results of DAML$_2$ are far better than those of DAML$_1$, which confirms that batch split in MLDG is very suitable for our DG task. DAML$_3$ outperforms DAML$_2$ by 2% to 5.1% in mAP accuracy, which proves that our proposed piecewise optimization strategy is more reasonable.

E. Parameter Analysis

In this section, we analyze the impacts of key hyperparameters on generality. The corresponding experimental results are shown in Fig. 8, where 8(a)∼(e) are the generalization experiments from Market1501 to DukeMTMC-reID dataset, and Fig. 8(f) is the generalization experiment from DukeMTMC-reID to Market1501 dataset. The source domain contains MSMT17 dataset in all the above experiments.

1) The Impact of Loss Coefficients λ_1, λ_2, and λ_3: The coefficients λ_1, λ_2, and λ_3 assign weights to losses $\mathcal{L}_{aad}$, $\mathcal{L}_{lde}$, and $\mathcal{R}$ respectively, representing their contribution. The impact of their different values on generality are shown in Fig. 8(a), (b), and (c). Here is how we determine their optimal values: first, we fix λ_2, and λ_3 to 1, and then try different values of λ_1 until we find the optimal one for generalization performance. Next, we fix λ_3 as 1, λ_1 as the obtained optimal value, and try different values of λ_2 to find the optimal one. After that, we get the optimal λ_3 in the same way. Finally, we get $\lambda_1 = 0.6$, $\lambda_2 = 0.5$, $\lambda_3 = 0.2$. At this point, the generalization performance is the best, Rank1=64.8%, mAP=43.3%. We conclude that the contribution of $\mathcal{L}_{aad}$ is the largest, followed by $\mathcal{L}_{lde}$, a little bit smaller than $\mathcal{L}_{aad}$, and $\mathcal{R}$ is the smallest.

2) The Impact of Dimension Reduction Ratio r: r is an important parameter of the bottleneck structure. It improves the generality of NL-CBAM by limiting model complexity. We choose different values of r in the experiment to explain its impact on domain generalization. The experimental results are shown in Fig. 8(d). When $r = 12$, the generalization performance of the model reaches the best. However, its impact is not very significant. When r changes from 1 to 20, the fluctuations of Rank1 and mAP are within 8.3% and 7.0% respectively.

3) Are Slot Numbers $s1$ and $s2$ Sensitive to Generality?: In theory, $s1$ and $s2$ are sensitive to generality, because they limit the number of labeled samples useful for pseudo label estimation and the number of reliable pseudo samples, respectively. Although the memory is scalable, the performance of the model can not be improved infinitely with the increase of capacity. Because as long as the memory can hold enough useful information, useless information tends to be cleared and replaced with useful information. This is proved by the generalization experiments on Market1501 and DukeMTMC-reID datasets. As can be seen from Fig. 8(e) and (f), when $s1 < 9000$ and $s2 < 5500$, as the memory capacity increases, the generality increases but the sensitivity decreases. When $s1 \geq 9000$ and $s2 \geq 5500$, the generality is basically not affected by $s1$ and $s2$. Here, $s1 = 9000$ and $s2 = 5500$ are empirical values, which may be slightly larger than the actual value because our RAM is sufficient. The values of $s1$ and $s2$ depend on the number of labeled and unlabeled images in the source domain.

F. Visualization

1) Attention Visualization: To qualitatively evaluate the superiority of NL-CBAM and AAN for DG tasks, we randomly select some test images from two target datasets and draw the heat maps, as shown in Fig. 9. These selected images cover all the misaligned cases in Fig. 1. By observing the heat maps obtained by our method and original CBAM method, we can intuitively see that our method focuses more on the pedestrian body and less on the occlusion or background. This improvement should be attributed to the bottleneck structure, non-local operation, and attention correction by AAN. It also proves that our designed NL-CBAM+AAN is more suitable for DG tasks than the original CBAM.

2) t-SNE Visualization: To further understand the effect of $\mathcal{L}_{lde}$ on the discrimination of features in M and M', we utilize t-SNE [46] to visualize the stored feature vectors of

M and M' by plotting them to the 2-dimension map. The t-SNE maps shown in Fig. 10 are obtained from the test dataset Market1501. From the changes Fig. 10(a)→(b) and (c)→(d), it can be observed that the features labeled as the same class in M and M' are gradually clustered together (from iteration 500 to iteration 1000), which proves that the discrimination of the features stored in M and M' increases with the training.

V. Conclusion

We propose a data assimilation network to tackle the domain generalization ReID task. We focus on three different types of images that are challenging for our DG task: style variants, misaligned and unlabeled images. Thereby we design the NL-CBAM, AAN and a progressive augmentation memory, and form a DAML procedure. Extensive experiments demonstrate that our proposed network achieves higher performance than the state-of-the-art DG methods, which proves that the idea of assimilating these three types of images to improve generality is very feasible. In the future, we will continue to mine the important information of unlabeled images to improve the generality of the model. Besides, we will extend this work to the fields of face recognition and vehicle re-identification, where the DG problem is prevalent.

References

[1] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 598–607.

[2] X. Zhang, J. Cao, C. Shen, and M. You, "Self-training with progressive augmentation for unsupervised cross-domain person re-identification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8221–8230.

[3] Y. Fu, Y. Wei, G. Wang, Y. Zhou, H. Shi, U. Uiuc, and T. Huang, "Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6111–6120.

[4] G. Chen, Y. Lu, J. Lu, and J. Zhou, "Deep credible metric learning for unsupervised domain adaptation person re-identification," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 643–659.

[5] J. Li and S. Zhang, "Joint visual and temporal consistency for unsupervised domain adaptive person re-identification," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 483–499.

[6] D. Mekhazni, A. Bhuiyan, G. Ekladious, and E. Granger, "Unsupervised domain adaptation in the dissimilarity space for person re-identification," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 159–174.

[7] C. Zhang, Y. Tang, Z. Zhang, D. Li, X. Yang, and W. Zhang, "Improving domain-adaptive person re-identification by dual-alignment learning with camera-aware image generation," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2020.

[8] Y. Zhai, S. Lu, Q. Ye, X. Shan, J. Chen, R. Ji, and Y. Tian, "Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9018–9027.

[9] S. Lin, C. T. Li, and A. C. Kot, "Multi-domain adversarial feature generalization for person re-identification," *IEEE Transactions on Image Processing*, vol. 30, pp. 1596–1607, 2021.

[10] X. Qian, Y. Fu, T. Xiang, Y. G. Jiang, and X. Xue, "Leader-based multi-scale attention deep architecture for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 371–385, 2020.

[11] J. Song, Y. Yang, Y. Song, T. Xiang, and T. M. Hospedales, "Generalizable person re-identification by domain-invariant mapping network," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 719–728.

[12] J. Jia, Q. Ruan, and T. M. Hospedales, "Frustratingly easy person re-identification: Generalizing person re-id in practice," *CoRR*, vol. abs/1905.03422, 2019. [Online]. Available: http://arxiv.org/abs/1905.03422

[13] X. Jin, C. Lan, W. Zeng, Z. Chen, and L. Zhang, "Style normalization and restitution for generalizable person re-identification," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3140–3149.

[14] L. X. T. Z. H. Z. H. W. H. A. Zijie Zhuang, Longhui Wei and Q. Tian, "Rethinking the distribution gap of person re-identification with camera-based batch normalization," in *ECCV*, 2020.

[15] S. Liao and L. Shao, "Interpretable and generalizable person re-identification with query-adaptive convolution and temporal lifting," *ECCV*, 2020.

[16] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[18] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 79–88.

[19] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[20] S. Lian, W. Jiang, and H. Hu, "Attention-aligned network for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2020.

[21] J. Lei, L. Niu, H. Fu, B. Peng, Q. Huang, and C. Hou, "Person re-identification by semantic region representation and topology constraint," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2453–2466, 2019.

[22] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato, "Hierarchical gaussian descriptor for person re-identification," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1363–1372.

[23] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 907–915.

[24] Y.-Z. S. T. M. H. Da Li, Yongxin Yang, "Learning to generalize: Meta-learning for domain generalization," in *The Association for the Advance of Artificial Intelligence (AAAI)*, April 2018.

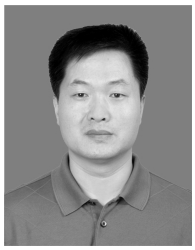
[25] S. Li, M. Shao, and Y. Fu, "Person re-identification by cross-view multi-level dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2963–2977, 2018.

[26] J. Z. Weihua Chen, Xiaotang Chen and K. Huang, "A multi-task deep network for person reidentification," in *AAAI*, 2017.

[27] S. Bai, X. Bai, and Q. Tian, "Scalable person re-identification on supervised smoothed manifold," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp.

3356–3365.
[28] S. G. Wei Li, Xiatian Zhu, "Person re-identification by deep joint learning of multi-loss classification," in *2017 IJCAI*, 2017.
[29] S. Bak and P. Carr, "One-shot metric learning for person re-identification," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1571–1580.
[30] L. Zhang, F. Liu, and D. Zhang, "Adversarial view confusion feature learning for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1490–1502, 2021.
[31] H. Tan, X. Liu, Y. Bian, H. Wang, and B. Yin, "Incomplete descriptor mining with elastic loss for person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021.
[32] J. Liu, Z.-J. Zha, Q. Tian, D. Liu, T. Yao, Q. Ling, and T. Mei, "Multi-scale triplet cnn for person re-identification," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 192–196.
[33] J. Liu, Z.-J. Zha, D. Chen, R. Hong, and M. Wang, "Adaptive transfer network for cross-domain person re-identification," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7195–7204.
[34] W. Li and X. Wang, "Locally aligned feature transforms across views," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3594–3601.
[35] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 152–159.
[36] B. W. L. L. Tong Xiao, Shuang Li and X. Wang, "End-to-end deep learning for person search," *arXiv preprint arXiv:1604.01850*, 2016.
[37] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.
[38] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3774–3782.
[39] D. Gray and H. Tao, "Viewpoint invariant pedestrian recognition with an ensemble of localized features," in *Computer Vision - ECCV 2008, European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings*, 2008, pp. 262–275.
[40] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof, "Person re-identification by descriptive and discriminative classification," in *Scandinavian Conference on Image Analysis*, 2011.
[41] C. C. Loy, C. Liu, and S. Gong, "Person re-identification by manifold ranking," in *2013 IEEE International Conference on Image Processing*, 2013, pp. 3567–3571.
[42] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by discriminative selection in video ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 12, pp. 2501–2514, 2016.
[43] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.
[44] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2921–2929.
[45] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1842–1850.
[46] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

Yixiu Liu received the B.E. degree from Northeastern University, China, in 2016. He is currently pursuing the Ph.D. degree at School of Information Science and Engineering, Northeastern University, Shenyang, China. He used to be a visiting scholar at the University of California, Riverside from Oct. 2018 to Oct. 2020. His research interests are in computer vision and machine learning.



Yunzhou Zhang received B.S. and M.S. degree in Mechanical and Electronic engineering from National University of Defense Technology, Changsha, China in 1997 and 2000, respectively. He received Ph.D. degree in pattern recognition and intelligent system from Northeastern University, Shenyang, China, in 2009. He is currently a professor with the Faculty of Robot Science and Engineering, Northeastern University, China. Now he leads the Cloud Robotics and Visual Perception Research Group.

Bir Bhanu (M82F95LF17) received the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, and the M.B.A. degree from the University of California at Irvine, Irvine, CA. He is currently the Bourns Presidential Chair in engineering, the Distinguished Professor of electrical and computer engineering, and the Founding Director of the Interdisciplinary Center for Research in Intelligent Systems and the Visualization and Intelligent Systems Laboratory, UCR. His research interests include computer vision, pattern recognition and data mining, machine learning, artificial intelligence, image processing, image and video database, graphics and visualization, robotics, human-computer interactions, and biological, medical, military, and intelligence applications. He was a Senior Honeywell Fellow with Honeywell Inc. He is a Fellow of IEEE, AAAS, IAPR, SPIE, and AIMBE.

Sonya Coleman (M11) received a BSc (Hons) in Mathematics, Statistics and Computing (first class) from the Ulster University, UK in 1999, and a PhD in Mathematics from the Ulster University, UK in 2003. She is a Professor and a leader in the Cognitive Robotics team of Intelligent Systems Research Centre. She is a Fellow of the Higher Education Academy. She has many publications in image processing, pattern recognition, computational intelligence and robotics. Her research has been supported by funding from various sources such as EPSRC, The Nuffield Foundation, The Leverhulme Trust and the European Commission. Additionally, she was co-investigator on the EU FP7 funded project RUBICON, the FP7 project VISUALISE and is currently co-investigator in the FP7 SLANDIAL project. She is also secretary of the Irish Pattern Recognition and Classification Society.

Dermot Kerr received a BSc(Hons) in Computing Science from the University of Ulster, UK in 2005, and a PhD in Computing and Engineering from the University of Ulster, UK in 2009. He is currently a research fellow in the School of Computing and Intelligent System at the University of Ulster, Magee. His current research interests are in mathematical image processing, feature detection, omnidirectional vision and robotics. Dr. Kerr is a member of the Irish Pattern Recognition and Classification Society.