Learning to Generalize Unseen Domains via Memory-based Multi-Source Meta-Learning for Person Re-Identification

Yuyang Zhao[1*], Zhun Zhong[2*], Fengxiang Yang[1], Zhiming Luo[1†], Yaojin Lin[4], Shaozi Li[1,3†], Nicu Sebe[2]

1 Department of Artificial Intelligence, School of Informatics, Xiamen University

2 Department of Information Engineering and Computer Science, University of Trento

3 Institute of Artificial Intelligence, Xiamen University
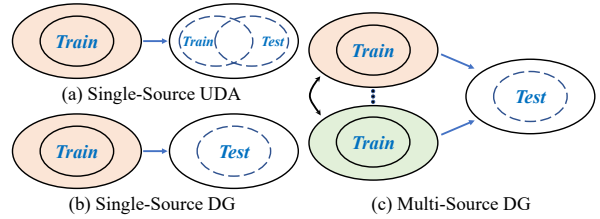
4 Minnan Normal University

Project: https://github.com/HeliosZhao/M3L

Abstract

*Recent advances in person re-identification (ReID) obtain impressive accuracy in the supervised and unsupervised learning settings. However, most of the existing methods need to train a new model for a new domain by accessing data. Due to public privacy, the new domain data are not always accessible, leading to a limited applicability of these methods. In this paper, we study the problem of multi-source domain generalization in ReID, which aims to learn a model that can perform well on unseen domains with only several labeled source domains. To address this problem, we propose the **M**emory-based **M**ulti-Source **M**eta-**L**earning (M^3L) framework to train a generalizable model for unseen domains. Specifically, a meta-learning strategy is introduced to simulate the train-test process of domain generalization for learning more generalizable models. To overcome the unstable meta-optimization caused by the parametric classifier, we propose a memory-based identification loss that is non-parametric and harmonizes with meta-learning. We also present a meta batch normalization layer (MetaBN) to diversify meta-test features, further establishing the advantage of meta-learning. Experiments demonstrate that our M^3L can effectively enhance the generalization ability of the model for unseen domains and can outperform the state-of-the-art methods on four large-scale ReID datasets.*

1. Introduction

Person re-identification (ReID) aims at matching persons of the same identity across multiple camera views. Recent works in ReID mainly focus on three settings, *i.e.*, fully-

*Equal contribution. This work was done when Yuyang Zhao (yuyangzhao98@gmail.com) was a visiting student at Xiamen University.

†Corresponding author: {zhiming.luo, szlig}@xmu.edu.cn

Setting	Source(s)			Target(s)	
	Multi	Images	Labels	Images	Labels
Fully-Supervised	✗	✓	✓	—	—
Fully-Unsupervised	✗	✓	✗	—	—
Single-Source UDA	✗	✓	✓	✓	✗
Single-Source DG	✗	✓	✓	✗	✗
Multi-Source DG	✓	✓	✓	✗	✗

Figure 1: Comparison of different settings in person ReID. Different background colors indicate different distributions, *i.e.*, domains. Solid/dashed ellipses denote data subset with/without labels. Domain generalization (DG) is designed to learn models for unseen domains, while other settings focus on learning models for specific domains. Compared to single-source DG, multi-source DG leverages knowledge from multiple labeled datasets, enforcing the model to learn more underlying patterns across domains.

supervised [45, 47, 55], fully-unsupervised [23, 24, 39] and unsupervised domain adaptive [8, 44, 52] ReID. Despite their good performance on a seen domain (*i.e.*, a domain with training data), most of them suffer from drastic performance decline on unseen domains. In real-world applications, the ReID systems will inevitably search persons in new scenes. Therefore, it is necessary to learn a model that has good generalization ability to unseen domains.

To meet this goal, domain generalization (DG) is a promising solution that aims to learn generalizable models with one or several labeled source domains. As shown in

Fig. 1, compared to other settings, DG does not require the access to target domains. Generally, DG can be divided into two categories, single-source DG [14, 22, 56] and multi-source DG [16, 33], according to the number of source domains. Recent works mainly focus on single-source DG where only one labeled source domain is available. However, a single domain provides limited training samples and scene information, restricting the improvement of single-source DG methods. In contrast, multi-source DG utilizes multiple datasets of different distributions, providing more training data that contain numerous variations and environmental factors. However, due to the strong compatibility of deep networks, directly aggregating all source domains together might lead the model to overfit on the domain bias, hampering the generalization ability of the model. Although we can sample balanced training data from all source domains during training to reduce the impact of domain bias, the above issue still remains.

In this paper, we study the multi-source DG and aim to enforce the model to learn discriminative features without domain bias so that the model can be generalized to unseen domains. To achieve this goal, this paper introduces a meta-learning strategy for multi-source DG, which simulates the train-test process of DG during model optimization. In our method, we dynamically divide the source domains into meta-train and meta-test sets at each iteration. The meta-train is regarded as source data, and the meta-test is regarded as "unseen" data. During training, we encourage the loss of meta-train samples to optimize the model towards a direction that can simultaneously improve the accuracy of meta-test samples. Nevertheless, meta-learning causes a problem for traditional parametric-based identification loss — unstable optimization. On the one hand, ReID datasets contain numerous IDs, so the number of classifier parameters will surge when multiple domains are used for training. On the other hand, the unified optimization of classifiers is unstable due to the asynchronous update by the high-order gradients of the meta-test. Consequently, we propose a memory-based identification loss, which uses a non-parametric memory to take full advantage of meta-learning while avoiding unstable optimization. We also introduce a meta batch normalization layer (MetaBN), which mixes meta-train knowledge with meta-test features to simulate the feature variations in different domains. Our full method is called **M**emory-based **M**ulti-Source **M**eta-**L**earning (M³L). Experiments on four large-scale ReID datasets demonstrate the effectiveness of our M³L when testing on unseen domains and show that our M³L can achieve state-of-the-art results.

Our contributions are summarized as follows:

- We propose a Multi-Source Meta-Learning framework for multi-source DG, which can simulate the train-test process of DG during training. Our method enables the model to learn domain-invariant representations and thus improves the generalization ability.

- We equip our framework with a memory-based module, which implements the identification loss in a non-parametric way and can prevent unstable optimization caused by traditional parametric manner during meta-optimization.

- We present MetaBN to generate diverse meta-test features, which can be directly injected into our meta-learning framework and obtain further improvement.

2. Related Work

Person Re-identification. Recently, supervised learning approaches [4, 34, 36, 40, 45, 47, 50, 54, 55] have achieved significant performance in person re-identification (ReID), relying on labeled training data. Considering the difficulties and complexities of annotations, unsupervised learning (USL) [6, 23, 24, 39] and unsupervised domain adaptation (UDA) [5, 8, 43, 44, 51, 52, 57] methods are proposed. UDA aims to utilize labeled source data and unlabeled target data to improve the model performance on the target domain. UDA methods mainly focus on generating pseudo-labels on target data [8, 44, 52] or transferring source images to the styles of the target domain for providing extra supervision during adaptation [5, 51, 57]. USL approaches learn discriminative features only from unlabeled target data, the mainstream [6, 23] of which is to train models with pseudo-labels obtained by clustering.

Domain Generalization. Although USL and UDA ReID methods show good performance, they still need to collect a large amount of target data for training models. In contrast, domain generalization (DG) has no access to any target domain data. By carefully designing, DG methods [3, 14, 17] can improve the model performance on unseen domains. Most existing DG methods focus on closed-set tasks [3, 15, 17, 26, 29], assuming that the target data have the same label space as the source data. Lately, several works [14, 22, 33, 56] were introduced to learn generalizable models for person ReID. SNR [14] disentangles identity-relevant and identity-irrelevant features and reconstructs more generalizable features. Liao *et al*. [22] propose a novel QAConv for calculating the similarity between samples, which can effectively improve ReID accuracy in unseen data but is inefficient during testing. DIMN [33] proposes a mapping subnet to match persons within a mini-batch and trains the model with data from one domain at each iteration. Song *et al*. [33] claim that DIMN uses meta-learning in the training stage. However, DIMN optimizes the model with the common training strategy, which is completely different from our meta-learning strategy.

Meta Learning. The concept of meta-learning [37] is learning to learn, and has been initially proposed in the
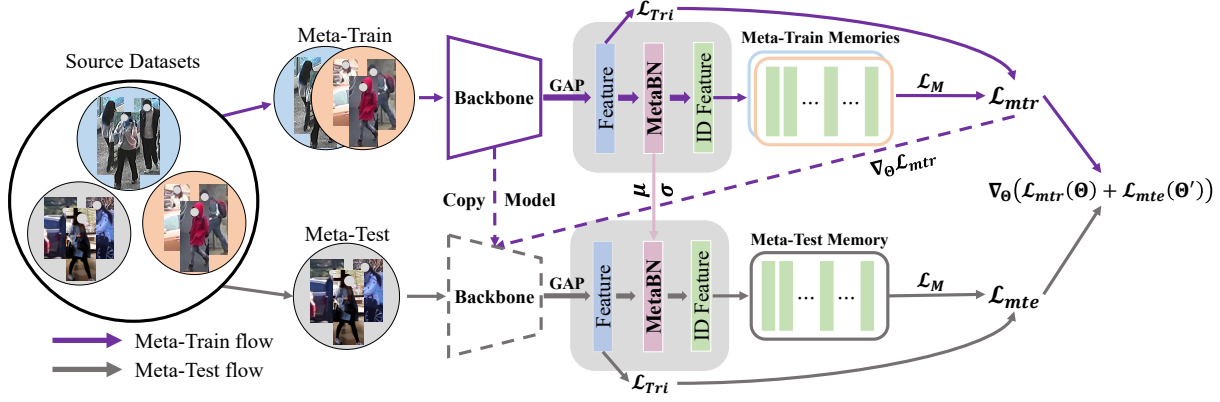
Figure 2: The framework of the proposed M^{3}L. During training, we are given several (three in this example) source domains. At each iteration, source domains are divided into one meta-test and two meta-train domains. In the meta-train stage, memory-based identification loss and triplet loss are calculated from meta-train data as the meta-train loss. In the meta-test stage, the original model is copied and then the copied model is updated with meta-train loss. We compute the meta-test loss on the updated model. In this stage, MetaBN is used to diversify the meta-test features. Finally, the combination of meta-train and meta-test losses is used to optimize the original model.

machine learning community. Recently, meta-learning has been applied to various deep-based applications, including model optimization [1, 19], few-shot learning [7, 32, 35, 38] and domain generalization [2, 10, 17, 18, 21]. MAML [7] and its variant Reptile [27] are proposed to learn a good initialization for fast adapting a model to a new task. Li *et al*. [17] first extend MAML [7] to closed-set DG. Latter, meta-learning was applied to closed-set DG [2, 18, 21] and open-set DG [10]. In this paper, we propose a memory-based meta-learning approach, which is tailor-made for multi-source DG in ReID.

3. Methodology

For multi-source domain generalization (DG) in person ReID, we are provided with N_S source domains $\mathcal{D}_S = \{\mathcal{D}_S^1, ..., \mathcal{D}_S^{N_S}\}$ in the training stage. The label spaces of the source domains are disjointed. The goal is to train a generalizable model with the source data. In the testing stage, the model is evaluated directly on a given unseen domain $\mathcal{D}_T$.

3.1. Overview

This paper designs a Memory-based Multi-source Meta-Learning (M^{3}L) framework for multi-source domain generalization (DG) in person ReID task. In our framework, we introduce a meta-learning strategy, which simulates the train-test process of DG during model optimization. Specifically, we dynamically split the source domains into meta-train and meta-test at each iteration. During training, we first copy the original model and update it with the loss from meta-train data. Then we use the updated model to compute the meta-test loss. The memory-based identification loss and triplet loss are adopted for effective meta-learning. We also inject a meta batch normalization layer (MetaBN)

into the network, which diversifies the meta-test features with meta-train distributions to further facilitate the effect of meta-learning. Finally, the combination of the meta-train and meta-test losses is used to update the original model towards a generalizable direction that performs well on meta-train and meta-test domains.

3.2. Meta-Learning for Multi-Source DG

We adopt the concept of "learning to learn" to simulate the train-test process of domain generalization during the model optimization. At each training iteration, we randomly divide N_S source domains into $N_S - 1$ domains as **meta-train** and the rest *one* domain as **meta-test**. The process of computing the meta-learning loss includes the meta-train and the meta-test stages.

In the meta-train stage, we calculate the meta-train loss $\mathcal{L}_{mtr}$ on the meta-train samples to optimize the model. *In the meta-test stage*, the optimized model is used to calculate the meta-test loss $\mathcal{L}_{mte}$ with the meta-test samples. Finally, the network is optimized by the combination of meta-train and meta-test losses, *i.e.*,

$$\arg\min_{\Theta} \mathcal{L}_{mtr}(\Theta) + \mathcal{L}_{mte}(\Theta^{'}), \tag{1}$$

where Θ denotes the parameters of the network, and $\Theta^{'}$ denotes the parameters of the model optimized by the $\mathcal{L}_{mtr}$. Note that, $\mathcal{L}_{mte}$ is only used to update Θ, the derivative of which is the high-order gradients on Θ.

Remark. In the proposed meta-learning objective, the meta-test loss encourages the loss of meta-train samples to optimize the model towards a direction that can improve the accuracy of meta-test samples. By iteratively enforcing the generalization process from meta-train domains to meta-test

domain, the model can avoid overfitting to domain bias and can learn domain-invariant representations that generalize well on unseen domains.

Next, we will introduce the loss functions used in meta-learning in Sec. 3.3, the MetaBN layer in Sec. 3.4 and detailed training procedure of meta-learning in Sec. 3.5.

3.3. Memory-based Identification Loss

Identification loss can effectively learn discriminative person representations in a classification manner. Commonly, a fully-connected layer is adopted as the classifier to produce the probabilities that are used for computing the cross-entropy loss. Although existing works [2, 31, 35] show the effectiveness of meta-learning in the classification task, the parametric classifier is inadequate in the context of ReID. This is because ReID is an open-set task, where different domains contain completely different identities and the number of identities in each domain is commonly large. In multi-source DG of ReID, we have two kinds of parametric classifier selections, one global FC classifier or N_S parallel FC classifiers for each domain, both of which will lead to problems during meta-learning.

For the global FC classifier (Fig. 3(a)), the dimension of the FC layer is the sum of all source identities. Different from closed-set tasks [2, 3], the global FC classifier contains a large number of parameters when trained with multiple person ReID datasets. This will lead to unstable optimization during the meta-learning. As for parallel FC classifiers in Fig. 3(b), although we can alleviate the parameter burden by only identifying persons within their own domain classifier, the number of parameters for all classifiers is still large. Moreover, during the meta-learning, the classifier of the meta-test domain is only updated by high-order gradients, which is asynchronous with the feature encoder. This optimization process is unequal and unstable, leading to an incomplete usage of meta-learning.

Taking all the above into consideration, inspired by [9, 42, 52, 53], we propose a memory-based identification loss for multi-source DG, which is non-parametric and suitable for both meta-learning and person ReID. As shown in Fig. 3(c), we maintain a feature memory for each domain, which contains the centroids of each identity. The similarities between features and memory centroids are used to compute the identification loss. The memory-based identification loss has two advantages to our meta-learning framework. *First,* the memory is a non-parametric classifier, which avoids the unstable optimization caused by a large number of parameters. *Second*, the asynchronous update between the feature encoder and memory has a slight influence on model training. This is because the memory is updated smoothly by a momentum instead of being updated by an optimizer. Thus, the memory is insensitive to the changes of the feature encoder caused by the last few train-
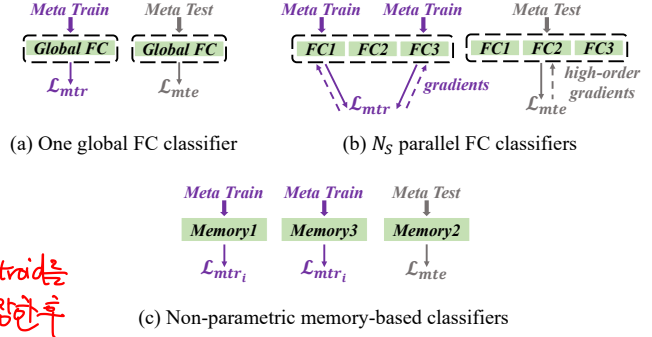

(a) One global FC classifier   (b) N_S parallel FC classifiers

(c) Non-parametric memory-based classifiers

Figure 3: Comparison of different classifiers. Layers within dashed rectangles are updated together by an optimizer.

ing iterations. In Sec. 4.4, we show that our meta-learning framework gains more improvements with the memory-based identification loss than with the FC-based identification loss. Next, we will introduce the memory-based identification loss in detail.

Memory Initialization. We maintain an individual memory for each source domain. For a source domain $\mathcal{D}_S^i$ with n_i identities, the memory $\mathcal{M}^i$ has n_i slots, where each slot saves the feature centroid of the corresponding identity. In initialization, we use the model to extract features for all samples of $\mathcal{D}_S^i$. Then, we initialize the centroid of each identity with a feature, which is averaged on the features of the corresponding identity. For simplicity, we omit the superscript of the domain index and introduce the memory updating and memory-identification loss for one domain.

Memory Updating. At each training iteration, we update the memory with the features in the current mini-batch. A centroid in the memory is updated through,

$$\mathcal{M}[k] \leftarrow m \cdot \mathcal{M}[k] + (1 - m) \cdot \frac{1}{|\mathcal{B}_k|} \sum_{x_i \in \mathcal{B}_k} f(x_i), \quad (2)$$

where $\mathcal{B}_k$ denotes the samples belonging to the kth identity and $|\mathcal{B}_k|$ denotes the number of samples for the kth identity in current mini-batch. $m \in [0, 1]$ controls the updating rate.

Memory-based identification loss. Given an embedding feature $f(x_i)$ from the forward propagation, we calculate the similarities between $f(x_i)$ and each centroid in the memory. The memory-based identification loss aims to classify $f(x_i)$ into its own identity, which is calculated by:

$$\mathcal{L}_M = -\log \frac{\exp\left(\mathcal{M}[i]^T f(x_i)/\tau\right)}{\sum_{k=1}^{n_i} \exp\left(\mathcal{M}[k]^T f(x_i)/\tau\right)}, \quad (3)$$

where τ is the temperature factor that controls the scale of distribution.

Triplet loss. We also use triplet loss [12] to train the model, which is formulated as,

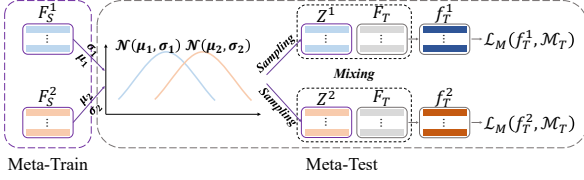$$\mathcal{L}_{Tri} = [d_p - d_n + \delta]_+, \quad (4)$$

4

Figure 4: Detailed architecture of MetaBN. MetaBN first utilizes the mean and variance of meta-train mini-batch features to construct Gaussian Distributions. Then, features sampled from these distributions are mixed with meta-test mini-batch features for generating new meta-test features.

where d_p is the Euclidean distance between an anchor feature and a hard positive feature, and d_n is the Euclidean distance between an anchor feature and a hard negative feature. δ is the margin of triplet loss and $[\cdot]_+$ refers to $max(\cdot, 0)$.

3.4. MetaBN →

In our meta-learning strategy, the meta-test loss is important for learning generalizable representations, since the meta-test plays the role of the "unseen" domain. Intuitively, if the meta-test examples are sampled from more diverse distributions, the model will be optimized to be more robust to variations and thus be more generalizable to unseen domains. To achieve this goal, we introduce MetaBN to generate more diverse meta-test features at the feature-level. As shown in Fig. 2, we replace the last batch normalization layer (BN) [13] in the network with MetaBN. During training, MetaBN utilizes the domain information from meta-train domains to inject domain-specific information into meta-test features. This process can diversify meta-test features, enabling the model to simulate more feature variations. The operation of MetaBN is illustrated in Fig. 4.

In the meta-train stage, for the ith meta-train domain, MetaBN normalizes the meta-train features as the traditional BN, and saves the mini-batch mean μ_i and mini-batch variance σ_i, which are used in the following meta-test stage.

In the meta-test stage, MetaBN uses the saved mean and variance to form $N_S - 1$ Gaussian Distributions. Note that, the generated distribution mainly reflects the high-level domain information instead of specific identity information. This is because each saved mean and variance is calculated over samples belonging to dozens of identities. Considering this factor, we sample features from these distributions and inject these domain-specific features into meta-test features.

Specifically, for the ith distribution, we sample one feature z_j^i for each meta-test feature:

$$z_j^i \sim \mathcal{N}\left(\mu_i, \sigma_i\right), \tag{5}$$

where $\mathcal{N}$ denotes Gaussian Distribution. By doing so, we obtain B (the batch size of meta-test features) sampled features, which are mixed with the original meta-test features

Algorithm 1: Training procedure of M^{3}L

Input: N_S source domains $\mathcal{D}_S = \{\mathcal{D}_S^1, ..., \mathcal{D}_S^{N_S}\}$
Init: Feature encoder F parametrized by Θ;
Inner loop learning rate α;
Outer loop learning rate β; Batch-size B;

1 **for** *iter in train_iters* **do**
2   Randomly select a domain as meta-test D_{mte};
3   Sample remaining domains as meta-train D_{mtr};
4   **Meta-Train:**
5   Sample B images X_S^k from each meta-train domain D_{mtr}^k;
6   Compute meta-train loss $\mathcal{L}_{mtr}$ (Eq.9);
7   Copy the original model and update the copied parameters Θ by Adam and inner loop learning rate α:
8      $\Theta' \leftarrow Adam(\nabla_\Theta \mathcal{L}_{mtr}, \Theta, \alpha)$;
9   **Meta-Test:**
10   Sample B images X_T from the meta-test domain D_{mte};
11   Diversify meta-test features with MetaBN;
12   Compute meta-test loss $\mathcal{L}_{mte}$ (Eq.10);
13   **Meta Optimization:**
14   Update the original model parameters Θ by Adam and outer loop learning rate β:
15   Gradient: $g \leftarrow \nabla_\Theta(\mathcal{L}_{mtr}(\Theta) + \mathcal{L}_{mte}(\Theta'))$;
16      $\Theta \leftarrow Adam(g, \Theta, \beta)$.
17 **end**

for generating new features F_T^i,

$$F_T^i = \lambda F_T + (1 - \lambda)Z^i, \tag{6}$$

where F_T denotes the original meta-test features. $Z^i = [z_0^i, z_1^i, \cdots, z_B^i]$ denotes B sampled features from the ith Gaussian Distribution. λ is the mixing coefficient, which is sampled from Beta Distribution, *i.e.*, $\lambda \sim \text{Beta}(1, 1)$.

Finally, the mixed features are normalized by batch normalization,

$$f_T^i = \gamma \frac{F_T^i - \mu_T^i}{\sqrt{{\sigma_T^i}^2 + \epsilon}} + \beta, \tag{7}$$

where μ_T^i and σ_T^i denote mini-batch mean and variance of F_T^i. γ and β denote the learnable parameters that scale and shift the normalized value.

3.5. Training procedure of M^{3}L

During training, N_S source domains are separated into $N_S - 1$ meta-train domains and *one* meta-test domain at each iteration. The model is optimized by the losses calculated in the meta-train and meta-test stages.

Meta-train. For each meta-train domain, the meta-train loss is a combination of memory-based identification (Eq.3)

5

Table 1: Statistics of Person ReID Benchmarks.

Benchmarks	# IDs	# images	# cameras
Market-1501 [46]	1,501	32,217	6
DukeMTMC-reID [48]	1,812	36,411	8
CUHK03 [20]	1,467	28,192	2
MSMT17 [41]	4,101	126,441	15

and triplet losses (Eq.4), *i.e.*, *i*th meta-train loss

$$\mathcal{L}_{mtr}^{i} = \mathcal{L}_{Tri}(X_S^i; \Theta) + \mathcal{L}_M(X_S^i, \mathcal{M}_S^i; \Theta), \quad (8)$$

where Θ denotes the parameters of the network. X_S^i and $\mathcal{M}_S^i$ denote the training samples and memory of the ith meta-train domain, respectively.

The total loss for meta-train is averaged over $N_S - 1$ meta-train domains, formulated as,

$$\mathcal{L}_{mtr} = \frac{1}{N_S - 1} \sum_{i=0}^{N_S - 1} \mathcal{L}_{mtr}^i. \quad (9)$$

Meta-test. In the meta-test stage, the meta-test domain is performed on the new parameters Θ', which is obtained by optimizing Θ with $\mathcal{L}_{mtr}$. With the MetaBN proposed in Sec. 3.4, we can obtain $N_S - 1$ mixed features for each meta-test sample. The average memory-based identification loss over these features is considered as the meta-test memory-based identification loss. The meta-test loss is:

$$\mathcal{L}_{mte} = \mathcal{L}_{Tri}(X_T; \Theta') + \frac{1}{N_S - 1} \sum_{k=0}^{N_S - 1} \mathcal{L}_M(f_T^k, \mathcal{M}_T; \Theta'), \quad (10)$$

where X_T denotes the meta-test samples and f_T^k denotes the kth mixed features generated by the MetaBN.

Meta Optimization. Finally, the model is optimized by the objective in Eq.1. The optimization procedure of our M³L is summarized in Alg. 1.

4. Experiments

4.1. Benchmarks and Evaluation Metrics

We conduct experiments on four large-scale person re-identification benchmarks: Market-1501 [46], DukeMTMC-reID [30, 48], CUHK03 [20, 49] and MSMT17 [41]. *For studying the multi-source DG*, we divide these four datasets into two parts: three domains as source domains for training and the other one as target domain for testing. The statistics of these four benchmarks are shown in Table 1. For simplicity, we denote Market-1501, DukeMTMC-reID, CUHK03, and MSMT17 as M, D, C, and MS in tables.

Note: In default, for CUHK03, we use the old protocol (CUHK03, 26,263 images of 1,367 IDs for training) as the source domain for training the model and the detected

subset of the new protocol (CUHK-NP [49]) as the target domain for testing; for MSMT17, we use the MSMT17_V2 for both training and testing. In Table 2, we also provide the results of using the detected subset of CUHK-NP (7,365 images of 767 IDs for training) and MSMT17_V1 for both training and testing, and we recommend using this setting in future studies.

The cumulative matching characteristic (CMC) at Rank-1 and mean average precision (mAP) are used to evaluate performance on the target testing set.

4.2. Implementation Details

We implement our method with two common backbones, *i.e.*, ResNet-50 [11] and IBN-Net50 [28]. Images are resized to 256×128 and the training batch size is set to 64. We use random flipping and random cropping for data augmentation. For the memory, the momentum coefficient m is set to 0.2 and the temperature factor τ is set to 0.05. The margin δ of triplet loss is 0.3. To optimize the model, we use Adam optimizer with a weight decay of 0.0005. The learning rate of inner loop α and outer loop β are initialized to 3.5×10^{-5} and increase linearly to 3.5×10^{-4} in the first 10 epochs. Then, α and β are decayed by 0.1 at the 30th epoch and 50th epoch. The total training stage takes 60 epochs.

Baseline. For the baseline, we directly train the model with the memory-based identification loss and triplet loss using the data of all the source domains. That is, the baseline does not apply the meta-learning strategy and MetaBN.

4.3. Comparison with State-of-the-Art methods

Since there is no multi-source DG method evaluating on large-scale datasets, we compare our method with state-of-the-art single-source DG methods, including OSNet-IBN [55], OSNet-AIN [56], SNR [14] and QAConv [22]. SNR [14] and QAConv [22] use the ResNet-50 as the backbone. OSNet-IBN [55] and OSNet-AIN [56] use their self-designed networks that have better performance than ResNet-50. When testing on Market-1501, DukeMTMC-reID, and CUHK03, the existing single-source DG methods utilize MSMT17 as the source domain for model training. They combine the train set and test set of MSMT17, which is denoted as Combined MSMT17 (Com-MS) in this paper. To verify that the effectiveness of our method is obtained by multi-source meta-learning instead of training with more IDs and images, we only use the training sets of the source domains for model training. For example, when using Market-1501 as the target domain, we train the model with the train sets of DukeMTMC-reID, CUHK03, and MSMT17, including 3,110 IDs and 75,406 images. The numbers of IDs and images are less than that of Combined MSMT17 (3,110 IDs *vs*. 4,101 IDs, and, 75,406 images *vs*. 126,441 images). To conduct a fair comparison, we reim-

Table 2: Comparison with State-of-the-Arts domain generalization methods on four large-scale person ReID benchmarks — Market-1501 (M), DukeMTMC-reID (D), CUHK03 (C) and MSMT17 (MS). The performance is evaluated quantitatively by mean average precision (mAP) and cumulative matching characteristic (CMC) at Rank-1 (R1).

Method	Source	IDs	Images	Market-1501 mAP	R1	Source	IDs	Images	DukeMTMC mAP	R1
OSNet-IBN [55]	Com-MS	4,101	126,441	37.2	66.5	Com-MS	4,101	126,441	45.6	67.4
OSNet-AIN [56]				43.3	70.1				52.7	71.1
SNR [14]				41.4	70.1				50.0	69.2
QAConv$_{50}$ [22]				43.1	72.6				52.6	69.4
QAConv$_{50}$ [22]*	MS+D+C	3,110	75,406	35.6	65.7	MS+M+C	3,159	71,820	47.1	66.1
M^3L (ResNet-50)				48.1	74.5				50.5	**69.4**
M^3L (IBN-Net50)				**50.2**	**75.9**				**51.1**	69.2
QAConv$_{50}$ [22]*†	MS+D +C-NP	2,510	56,508	39.5	68.6	MS+M +C-NP	2,559	52,922	43.4	64.9
M^3L (ResNet-50)†				51.1	76.5				48.2	67.1
M^3L (IBN-Net50)†				**52.5**	**78.3**				**48.8**	**67.2**

Method	Source	IDs	Images	CUHK-NP mAP	R1	Source	IDs	Images	MSMT17 mAP	R1
QAConv$_{50}$ [22]	Com-MS	4,101	126,441	22.6	25.3	D	702	16,522	8.9	29.0
QAConv$_{50}$ [22]*	MS+D+M	2,494	62,079	21.0	23.5	D+M+C	2,820	55,748	7.5	24.3
M^3L (ResNet-50)				29.9	30.7				12.9	33.0
M^3L (IBN-Net50)				**32.1**	**33.1**				**14.7**	**36.9**
QAConv$_{50}$ [22]*†	MS+D+M	2,494	62,079	19.2	22.9	D+M +C-NP	2,220	36,823	10.0	29.9
M^3L (ResNet-50)†				30.9	**31.9**				13.1	32.0
M^3L (IBN-Net50)†				**31.4**	31.6				**15.4**	**37.1**

* We reimplement this work based on the authors' code on Github with the same source datasets as us.

† Model is trained and tested on the detected subset of CUHK-NP and MSMT17_V1.

plement recent published QAConv [22] with the same training data as us. Comparison results are reported in Table 2.

Results on Market-1501 and DukeMTMC-reID. From Table 2, we can make the following observations. First, when using Combined MSMT17 as the source data, OSNet-AIN [56] and QAConv [22] achieve the best results on both Market-1501 and DukeMTMC-reID. Second, compared to single-source DG methods that use more training data (Combined MSMT17), our M^3L outperforms them by a large margin on Market-1501 and achieves comparable results with them on DukeMTMC-reID. Specifically, when testing on Market-1501, with the same backbone, our M^3L surpasses SNR [14] by 6.7% in mAP and 4.4% in Rank-1 accuracy. Third, when training with multiple source domains, with the same backbone, our M^3L produces significantly higher results than QAConv$_{50}$. Specifically, our M^3L is higher than QAConv$_{50}$ by 12.5% in mAP for Market-1501 and by 3.4% in mAP for DukeMTMC-reID. This demonstrates the superiority of our method over the method that considers all the source domains as one domain. Fourth, when using the IBN-Net50 as the backbone, our M^3L can achieve better mAP than using ResNet-50.

Results on CUHK03 and MSMT17. There is only one method (QAConv [22]) evaluated on CUHK03 and MSMT17. When testing on MSMT17, QAConv [22] uses DukeMTMC-reID as the source data. Clearly, our M^3L achieves higher results than QAConv [22] on both datasets,

no matter how many source domains QAConv is trained with. We also find that both our M^3L and QAConv produce poor results on CUHK03 and MSMT17, indicating there is still a large room for generalizable models in DG.

4.4. Ablation Studies

Effectiveness of Meta-Learning. To investigate the effectiveness of the proposed meta-learning strategy, we conduct ablation studies in Table 3. Clearly, the model trained with the proposed meta-learning strategy consistently improves the results with different backbones. Specifically, with ResNet-50, adding meta-learning optimization increases the baseline by 5.3% in Rank-1 accuracy on Market-1501 and by 3.7% in Rank-1 accuracy on CUHK03. With IBN-Net50 backbone, meta-learning strategy gains 5.4% and 2.8% improvement in mAP on Market-1501 and CUHK03, respectively. This demonstrates that by simulating the train-test process during training, the meta-learning strategy helps the model to learn domain-invariant representations that can perform well on unseen domains.

Effectiveness of MetaBN. As shown in Table 3, plugging MetaBN into the meta-learning-based model further improves the generalization ability. For ResNet-50 backbone, MetaBN improves the meta-optimized model by 1.3% and 1.6% in Rank-1 accuracy on Market-1501 and CUHK03. For IBN-Net50, we can observe similar improvements. The results validate that diversifying meta-test fea-

Table 3: Ablation studies on meta-learning strategy and MetaBN. Models are trained with the other three datasets except the target dataset. "Meta": training with meta-learning strategy; "MetaBN": training with MetaBN.

Backbone	Meta	MetaBN	MS+D+C→M mAP	R1	MS+D+M→C mAP	R1
ResNet-50	✗	✗	41.1	67.9	25.7	25.4
	✓	✗	47.4	73.2	29.1	29.1
	✓	✓	**48.1**	**74.5**	**29.9**	**30.7**
IBN-Net50	✗	✗	43.6	71.1	28.2	29.4
	✓	✗	49.0	75.0	31.0	31.8
	✓	✓	**50.2**	**75.9**	**32.1**	**33.1**

Table 4: Comparison of loss function components. $\mathcal{L}_M$ and $\mathcal{L}_{Tri}$ denote memory-based identification loss and triplet loss. Experiments are conducted with ResNet-50.

$\mathcal{L}_M$	$\mathcal{L}_{Tri}$	MS+D+C→M mAP	R1	MS+D+M→C mAP	R1
✓	✗	47.8	73.6	**29.9**	30.3
✗	✓	35.1	59.7	20.9	19.9
✓	✓	**48.1**	**74.5**	**29.9**	**30.7**

Table 5: Comparison of different classifiers. $\mathcal{L}_M$ denotes memory-based identification loss. $\mathcal{L}_{FCG}$ and $\mathcal{L}_{FCP}$ denote traditional identification loss with global classifier and parallel classifiers. "Meta" denotes training with the meta-learning strategy and MetaBN. Numbers in parentheses denote performance improvement gained by "Meta". Experiments are conducted with ResNet-50.

Loss	Meta	MS+D+C→M mAP	R1	MS+D+M→C mAP	R1
$\mathcal{L}_{FCG}$	✗	37.7	67.0	21.2	20.9
	✓	39.7 (2.0)	68.3 (1.3)	21.2 (0.0)	21.9 (1.0)
$\mathcal{L}_{FCP}$	✗	37.7	67.0	21.2	20.9
	✓	40.9 (3.2)	69.3 (2.3)	23.9 (2.7)	24.3 (3.4)
$\mathcal{L}_M$	✗	41.1	67.9	25.7	25.4
	✓	48.1 (**7.0**)	74.5 (**6.6**)	29.9 (**4.2**)	30.7 (**5.3**)

Table 6: Comparison of training with different source domains. Experiments are conducted with ResNet-50.

Sources	Market-1501 mAP	R1	Sources	CUHK03 mAP	R1
MS+D	38.5	66.2	MS+D	21.9	23.7
MS+C	39.8	65.8	MS+M	27.1	27.8
MS+D+C	**48.1**	**74.5**	MS+D+M	**29.9**	**30.7**

tures by MetaBN is able to help the model to learn more generalizable representations for unseen domains.

Loss function components. We conduct experiments to evaluate the impact of the memory-based identification loss and triplet loss. Results in Table 4 show that the memory-based identification loss $\mathcal{L}_M$ is the predominant supervision for training a generalizable model and additionally adding the triplet loss $\mathcal{L}_{Tri}$ can slightly improve the performance.

Comparison of different classifiers. In Table 5, we compare different types of identification classifiers. We have the following observations. First, compared with the two parametric classifiers, our proposed non-parametric classifier gains higher improvement with the meta-learning strategy. Second, when directly training with multi-source data without the meta-learning, the model trained with memory-based identification loss achieves higher results. These two observations demonstrate that the proposed memory-based identification loss is suitable for multi-source DG and our meta-learning strategy.

Effectiveness of Multi-Source. Table 6 shows the comparison between two-source DG and multi-source DG. Despite bringing more domain bias, training with more source domains consistently produces higher results when testing on an unseen domain. This demonstrates the significance of studying multi-source DG.

4.5. Visualization

To better understand the effectiveness of our approach, we visualize the t-SNE [25] distributions of the features on the four datasets for different models, *i.e.*, ImageNet pretrained model, baseline, and M³L. Results are shown in Fig. 5. As shown in Fig. 5(a), without training, distributions
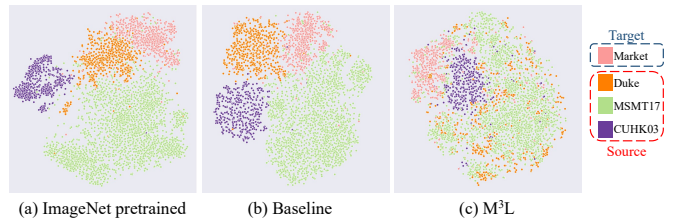

(a) ImageNet pretrained   (b) Baseline   (c) M³L

Figure 5: Visual distributions of four person ReID benchmarks. The distributions are obtained from inference features of (a) ImageNet pretrained model, (b) Baseline, and (c) M³L. All of the models are trained with ResNet-50, and the dimension of inference features is reduced by t-SNE [25].

vary by domains: (1) MSMT17 is the largest dataset that contains images in a variety of situations; (2) DukeMTMC-reID and Market-1501 are closely related to MSMT17 and each other; (3) CUHK03 has a relatively more distinct distribution compared with the others. Fig. 5(b) and Fig. 5(c) show the visual distributions of the four datasets after training. The model is trained on DukeMTMC-reID, CUHK03, and MSMT17, and tested on Market-1501. Comparing Fig. 5(b) with Fig. 5(c), we observe that the features from the source and target domains of M³L (Fig. 5(c)) are clustered more compactly, indicating that M³L leads the model to learn more generic and domain-agnostic representations.

5. Conclusion

In this paper, we propose a Memory-based Multi-source Meta-Learning (M^3L) framework for multi-source domain generalization (DG) in person ReID. The proposed meta-learning strategy enables the model to simulate the train-test process of DG during training, which can efficiently improve the generalization ability of the model on unseen domains. Besides, we introduce a memory-based module and MetaBN to take full advantage of meta-learning and obtain further improvement. Extensive experiments demonstrate the effectiveness of our framework for training a generalizable ReID model. Our method achieves state-of-the-art generalization results on four large-scale benchmarks.

Appendix

A. Visualization of the target domain

To better understand the advantage of the meta-learning strategy, we visualize the distributions of the inference features of the target domain (Market-1501 testing set) in Fig. A. Both baseline and M^3L are trained with DukeMTMC-reID, CUHK03, and MSMT17, and the inference features are obtained by 7 persons in the Market-1501 testing set. We use t-SNE [25] to reduce the features into a 2-D space. Different colors denote different identities. As shown in Fig. A, compared with the baseline, our M^3L pushes the features of the same identity more compact and pull the features of different identities more discriminating. This suggests that the proposed M^3L leads the model to learn more generalizable representations that can perform well on unseen domains.

Table A: Results on different number of training IDs (the improvement in red).

Loss	Meta	#training IDs (D+MS+C→M) Rank-1 accuracy				
		50	100	500	1,000	3,110
$\mathcal{L}_{FCG}$	×	27.1	35.2	48.5	54.5	67.0
	✓	**30.2** (3.1)	**37.3** (2.1)	49.0 (0.5)	55.3 (0.8)	68.3 (1.3)
$\mathcal{L}_{FCP}$	×	27.1	35.2	48.5	54.5	67.0
	✓	28.7 (1.6)	36.8 (1.6)	49.4 (0.9)	55.9 (1.4)	69.3 (2.3)
$\mathcal{L}_{M}$	×	27.6	34.6	51.6	59.6	67.9
	✓	28.0 (0.4)	35.3 (0.7)	**53.8** (2.2)	**63.4** (3.8)	**74.5** (6.6)

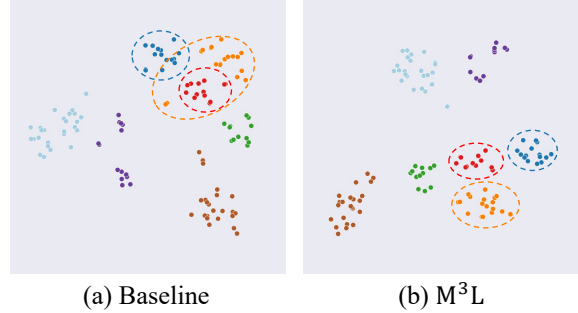(a) Baseline   (b) M^3L

Figure A: t-SNE [25] visualization of 7 persons in the unseen target dataset (Market-1501 testing set). The color indicates the identity. Results are evaluated on (a) baseline and (b) M^3L, both of which are trained with ResNet-50.

B. Detailed comparison of different classifiers

Meta-learning is effective with the FC-based classifiers in many tasks, *e.g.,* few-shot learning [7, 17]. However, we found that the advantage of meta-learning with the FC-based classifiers will be degraded when the number of classes (IDs) is large. In Table A, we compare the results of three kinds of classifiers with different number of training IDs (#training IDs). With fewer IDs, the FC-based classifiers achieve higher improvement. However, with the increase of IDs, the memory-based classifier gains higher improvement. Hence, we conclude that the FC-based classifiers are not suitable for meta-learning when the the number of classes is large. Thus, the large number of IDs in ReID leads the FC-based classifiers to produce inferior improvements than the memory-based classifier with meta-learning.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. 3

[2] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *NeurIPS*, 2018. 3, 4

[3] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *ECCV*, 2020. 2, 4

[4] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abd-net: Attentive but diverse person re-identification. In *CVPR*, 2019. 2

[5] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Instance-guided context rendering for cross-domain person re-identification. In *ICCV*, 2019. 2

[6] Hehe Fan, Liang Zheng, Chenggang Yan, and Yi Yang. Unsupervised person re-identification: Clustering and fine-tuning. *TOMM*, 2018. 2

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 3, 9

[8] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S Huang. Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *ICCV*, 2019. 1, 2

[9] Yixiao Ge, Dapeng Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In *NeurIPS*, 2020. 4

[10] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *CVPR*, 2020. 3

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[12] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017. 4

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 5

[14] Xin Jin, Cuiling Lan, Wenjun Zeng, Zhibo Chen, and Li Zhang. Style normalization and restitution for generalizable person re-identification. In *CVPR*, 2020. 2, 6, 7

[15] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012. 2

[16] Devinder Kumar, Parthipan Siva, Paul Marchwica, and Alexander Wong. Fairest of them all: Establishing a strong baseline for cross-domain person reid. *arXiv:1907.12016*, 2019. 2

[17] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *AAAI*, 2018. 2, 3, 9

[18] Da Li, Jianshu Zhang, Yongxin Yang, Cong Liu, Yi-Zhe Song, and Timothy M. Hospedales. Episodic training for domain generalization. In *ICCV*, 2019. 3

[19] Ke Li and Jitendra Malik. Learning to optimize. In *ICLR*, 2017. 3

[20] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 6

[21] Yiying Li, Yongxin Yang, Wei Zhou, and Timothy M Hospedales. Feature-critic networks for heterogeneous domain generalization. In *ICML*, 2019. 3

[22] Shengcai Liao and Ling Shao. Interpretable and generalizable person re-identification with query-adaptive convolution and temporal lifting. In *ECCV*, 2020. 2, 6, 7

[23] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *AAAI*, 2019. 1, 2

[24] Yutian Lin, Lingxi Xie, Yu Wu, Chenggang Yan, and Qi Tian. Unsupervised person re-identification via softened similarity learning. In *CVPR*, 2020. 1, 2

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008. 8, 9

[26] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, 2013. 2

[27] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv:1803.02999*, 2018. 3

[28] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *ECCV*, 2018. 6

[29] Fengchun Qiao, Long Zhao, and Xi Peng. Learning to learn single domain generalization. In *CVPR*, 2020. 2

[30] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCVW*, 2016. 6

[31] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 4

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 3

[33] Jifei Song, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Generalizable person re-identification by domain-invariant mapping network. In *CVPR*, 2019. 2

[34] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *ECCV*, 2018. 2

[35] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *CVPR*, 2019. 3, 4

[36] Chiat-Pin Tay, Sharmili Roy, and Kim-Hui Yap. Aanet: Attribute attention network for person re-identifications. In *CVPR*, 2019. 2

[37] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*. Springer, 1998. 2

[38] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016. 3

[39] Dongkai Wang and Shiliang Zhang. Unsupervised person re-identification via multi-label classification. In *CVPR*, 2020. 1, 2

[40] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *ACM MM*, 2018. 2

[41] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018. 6

[42] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017. 4

[43] Yunpeng Zhai, Shijian Lu, Qixiang Ye, Xuebo Shan, Jie Chen, Rongrong Ji, and Yonghong Tian. Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *CVPR*, 2020. 2

[44] Yunpeng Zhai, Qixiang Ye, Shijian Lu, Mengxi Jia, Rongrong Ji, and Yonghong Tian. Multiple expert brainstorming for domain adaptive person re-identification. In *ECCV*, 2020. 1, 2

[45] Zhizheng Zhang, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Relation-aware global attention for person re-identification. In *CVPR*, 2020. 1, 2

[46] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *CVPR*, 2015. 6

[47] Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, and Jan Kautz. Joint discriminative and generative learning for person re-identification. In *CVPR*, 2019. 1, 2

[48] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *ICCV*, 2017. 6

[49] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *CVPR*, 2017. 6

[50] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 2

[51] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *ECCV*, 2018. 2

[52] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *CVPR*, 2019. 1, 2, 4

[53] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Learning to adapt invariance in memory for person re-identification. *TPAMI*, 2020. 4

[54] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camstyle: A novel data augmentation method for person re-identification. *TIP*, 2019. 2

[55] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *ICCV*, 2019. 1, 2, 6, 7

[56] Kaiyang Zhou, Xiatian Zhu, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning generalisable omni-scale representations for person re-identification. *arXiv:1910.06827*, 2019. 2, 6, 7

[57] Yang Zou, Xiaodong Yang, Zhiding Yu, BVK Kumar, and Jan Kautz. Joint disentangling and adaptation for cross-domain person re-identification. In *ECCV*, 2020. 2