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

Yuyang Zhao\textsuperscript{1}, Zhun Zhong\textsuperscript{2}, Fengxiang Yang\textsuperscript{1}, Zhiming Luo\textsuperscript{1,}\textsuperscript{†}, Yaojin Lin\textsuperscript{1}, Shaozi Li\textsuperscript{1,}\textsuperscript{3,}\textsuperscript{†}, Nicu Sebe\textsuperscript{2}

\textsuperscript{1}Department of Artificial Intelligence, School of Informatics, Xiamen University
\textsuperscript{2}Department of Information Engineering and Computer Science, University of Trento
\textsuperscript{3}Institute of Artificial Intelligence, Xiamen University
\textsuperscript{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 Memory-based Multi-Source Meta-Learning (M\textsuperscript{3}L) 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 in ReID, which aims to learn a model that can perform well on unseen domains with only several labeled source domains. 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\textsuperscript{3}L 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-supervised \cite{45, 47, 55}, fully-unsupervised \cite{23, 24, 39} and unsupervised domain adaptive \cite{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 [16, 33] and multi-source DG [14, 22, 56] 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 to 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 Memory-based Multi-Source Meta-Learning (M^3L). Experiments on four large-scale ReID datasets demonstrate the effectiveness of our M^3L when testing on unseen domains and show that our M^3L 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 minibatch 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
machine learning community. Recently, meta-learning has been applied to various deep-based applications, including model optimization \cite{1,19}, few-shot learning \cite{7,32,35,38} and domain generalization \cite{2,10,17,18,21}. MAML \cite{7} and its variant Reptile \cite{27} are proposed to learn a good initialization for fast adapting a model to a new task. Li et al. \cite{17} first extend MAML \cite{7} to closed-set DG. Latter, meta-learning was applied to closed-set DG \cite{2,18,21} and open-set DG \cite{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 $D_S = \{D_S^1, ..., D_S^{N_S}\}$ 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 $D_T$.

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.,

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

where $\Theta$ 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 $\Theta$, the derivative of which is the high-order gradients on $\Theta$.

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 training 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 \( D_S^i \) 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 \( D_S^i \). 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,

\[
\mathcal{M}[k] \leftarrow m \cdot \mathcal{M}[k] + (1-m) \cdot \frac{1}{|B_k|} \sum_{x_i \in B_k} f(x_i),
\]

where \( B_k \) denotes the samples belonging to the \( k \)th identity and \( |B_k| \) denotes the number of samples for the \( k \)th 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:

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

where \( \tau \) 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,

\[
\mathcal{L}_T r i = [d_p - d_n + \delta]_+,
\]
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. $\delta$ is the margin of triplet loss and $[.]_+$ 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 $i$th meta-train domain, MetaBN normalizes the meta-train features as the traditional BN, and saves the mini-batch mean $\mu_i$ and mini-batch variance $\sigma_i$, 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 $i$th distribution, we sample one feature $z_j^i$ for each meta-test feature:

$$z_j^i \sim \mathcal{N}(\mu_i, \sigma_i),$$  

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 for generating new features $F_T^i$,

$$F_T^i = \lambda F_T + (1 - \lambda)Z^i,$$  

where $F_T$ denotes the original meta-test features. $Z^i = [z_0^i, z_1^i, \ldots, z_B^i]$ denotes $B$ sampled features from the $i$th Gaussian Distribution. $\lambda$ 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,

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

where $\mu_T^i$ and $\sigma_T^i$ denote mini-batch mean and variance of $F_T^i$. $\gamma$ and $\beta$ denote the learnable parameters that scale and shift the normalized value.

### 3.5. Training procedure of M3L

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)
and triplet losses (Eq.4), i.e.,
\[ L_{mtr} = L_{Tr}(X^i_S; \Theta) + L_M(X^i_S, M^i_S; \Theta), \] (8)

where \( \Theta \) denotes the parameters of the network. \( X^i_S \) and \( M^i_S \) denote the training samples and memory of the \( i \)th meta-train domain, respectively.

The total loss for meta-train is averaged over \( N_S - 1 \) meta-train domains, formulated as,
\[ L_{mtr} = \frac{1}{N_S - 1} \sum_{i=0}^{N_S-1} L_{mtr}^i. \] (9)

**Meta-test.** In the meta-test stage, the meta-test domain is performed on the new parameters \( \Theta' \), which is obtained by optimizing \( \Theta \) with \( L_{mtr} \). 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:
\[ L_{mte} = L_{Tr}(X_T; \Theta') + \frac{1}{N_S - 1} \sum_{k=0}^{N_S-1} L_M(f^k_T, M_T; \Theta'), \] (10)

where \( X_T \) denotes the meta-test samples and \( f^k_T \) denotes the \( k \)th mixed features generated by the MetaBN.

**Meta Optimization.** Finally, the model is optimized by the objective in Eq.1. The optimization procedure of our M^3L 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. For CUHK03, we adopt CUHK-NP [49] detected subset for evaluation. The statistics of these four benchmarks are shown in Table 1.

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

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 \( \tau \) is set to 0.05. The margin \( \delta \) 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 \( \alpha \) and outer loop \( \beta \) are initialized to \( 3.5 \times 10^{-5} \) and increase linearly to \( 3.5 \times 10^{-4} \) in the first 10 epochs. Then, \( \alpha \) and \( \beta \) 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 reimplement 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 M3L 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 M3L produces significantly higher results than QAConv. Specifically, our M3L is higher than QAConv 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 M3L 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 M3L 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 M3L 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 simulat-
Table 4: Comparison of loss function components. $L_M$ and $L_{TVI}$ denote memory-based identification loss and triplet loss. Experiments are conducted with ResNet-50.

<table>
<thead>
<tr>
<th>Loss</th>
<th>Meta</th>
<th>$L_{TVI}$</th>
<th>$L_{FPC}$</th>
<th>$L_{FCC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>×</td>
<td>×</td>
<td>$M+S+D+C$</td>
<td>$M+S+D+M$</td>
</tr>
<tr>
<td>$L_M$</td>
<td>✓</td>
<td>✓</td>
<td>47.8</td>
<td>29.9</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>35.1</td>
<td>20.9</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>48.1</td>
<td>29.9</td>
</tr>
</tbody>
</table>

Table 5: Comparison of different classifiers. $L_M$ denotes memory-based identification loss. $L_{FCC}$ and $L_{FPC}$ 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.

<table>
<thead>
<tr>
<th>Sources</th>
<th>Market-1501 mAP</th>
<th>Source</th>
<th>CUHK03 mAP</th>
<th>R1</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS+D</td>
<td>38.5</td>
<td>MS+D</td>
<td>21.9</td>
<td>23.7</td>
</tr>
<tr>
<td>MS+C</td>
<td>39.8</td>
<td>MS+M</td>
<td>27.1</td>
<td>27.8</td>
</tr>
<tr>
<td>MS+D+C</td>
<td>48.1</td>
<td>MS+D+M</td>
<td>29.9</td>
<td>30.7</td>
</tr>
</tbody>
</table>

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) M^3L. All of the models are trained with ResNet-50, and the dimension of inference features is reduced by t-SNE [25].

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.

Acknowledgements This work is supported by the National Nature Science Foundation of China (No. 61876159, 61806172, 61662024, 62076116 and U1705286); the China Postdoctoral Science Foundation Grant (No. 2019M652257); the Fundamental Research Funds for the Central Universities (Xiamen University, No. 20720200030); the EU H2020 SPRING No. 871245 and AI4Media No. 951911 projects; and the Italy-China collaboration project TALENT: 2018YFE0118400.
References


[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
[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