Generalized Person Re-identification by Locating and Eliminating Domain-Sensitive Features

Wendong Wang[1], Fengxiang Yang[1], Zhiming Luo[1*], and Shaozi Li[1]

[1]Department of Artificial Intelligence, Xiamen University
{wendongwang, yangfx}@stu.xmu.edu.cn,
{zhiming.luo, szlig}@xmu.edu.cn

Abstract. In this paper, we study the problem of domain generalization for person re-identification (re-ID), which adopts training data from multiple domains to learn a re-ID model that can be directly deployed to unseen target domains without further fine-tuning. One promising idea is removing the subsets of features that are not beneficial to the generalization of models. This can be achieved by muting the subset features that correspond to high back-propagated gradients as these subsets are easy for the model to overfit. But this method ignores the interaction of multiple domains. Therefore, we propose a novel method to solve this problem by comparing the gradients from two different training schemes. One of the training schemes discriminates input data from their corresponding domain to obtain back-propagated temporary gradients in the intermediate features. At the same time, another scheme discriminates input data from all domains to obtain the temporary gradients. By comparing the temporary gradient between the two schemes, we can identify the domain-generalizable subset features from those domain-specific subset features. We thus mute them in the subsequent training process to enforce the model to learn domain-generalizable information and improve its generalization. Extensive experiments on four large-scale re-ID benchmarks have verified the effectiveness of our method. Code is available at https://github.com/Ssd111/LEDF.git.

1 Introduction

Person re-identification (re-ID) aims at retrieving the target person in a non-overlapped camera system, which is a crucial technique for public security. Although person re-ID algorithms have achieved remarkable success [28, 30, 46, 33, 35, 37] with the help of deep neural networks [7, 9], most of them still suffer from the problem of generalization. Concretely, the re-ID model trained on a labeled source domain may perform well but fails to achieve high accuracy when transferred to another unseen target domain. Moreover, optimizing the re-ID model requires a large number of labeled samples, which is not an economical way in real-world applications. Recent advances adopt unsupervised domain adaptation [42, 26, 1, 4, 32] or fully unsupervised re-ID [34, 5, 3, 36] to address the

*: Corresponding Author.

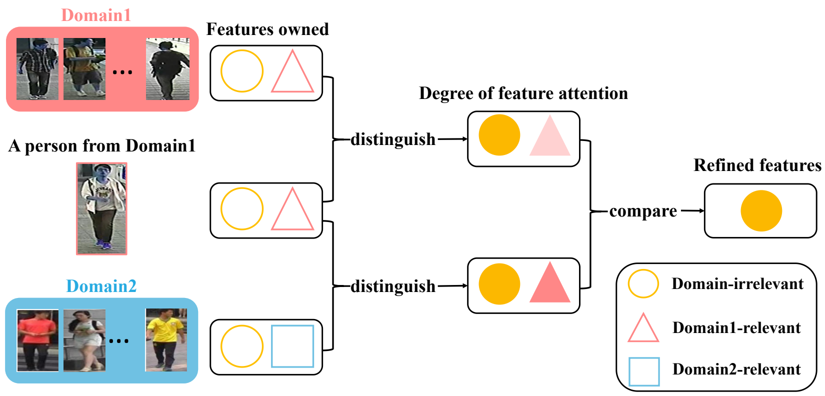

Fig. 1. Illustration of motivation and our idea. We use different shapes to represent different features, and the depth of color to indicate the degree of feature attention. We compare the degree of feature attention to eliminate domain-relevant features.

problem efficiently with the unlabeled data. However, these methods may not be applicable to the unseen target domains with strict privacy-preserving regulations since training data is not available. That's the domain generalization (DG) problem, which requires the model to perform well in the unseen target domains. Therefore, we consider studying the domain generalized re-ID algorithm in this paper. Prevailing generalization methods can be roughly categorized into three types: (1) data augmentation [44, 45, 16, 10, 17], (2) domain-invariant feature learning [22, 15, 13, 23, 20, 12], and (3) gradient calibration [14, 38, 27, 21]. Specifically, (1) improves the generalization by generating more diverse samples for optimization, (2) tries to use distributional metrics to narrow down the domain gaps of training domains for better representation learning, and (3) calibrates the training gradients for better learning of models. Recently, some methods like RSC [11] consider fine-grained constraints in representation learning. Specifically, RSC identifies the most predictive subsets of features through their associated gradients. The subsets with higher gradients are more likely to contain information that can mislead the model to overfit on the specific source domain and thus should be muted during the optimization. Although RSC has achieved great success, it ignores the interaction among source domains, hindering the further improvement in domain generalized re-ID.

Inspired by the idea of finding usable subsets of features during the optimization, we try to solve the DG problem in re-ID by locating and eliminating the domain-sensitive features (LEDF). This is achieved by comparing the gradients from two different training schemes for the same inputs, which fully considers the interaction among source domains. We compare two types of training schemes to locate domain-sensitive subsets. They are ① discriminating the sample from IDs of the same domain (domain-specific scheme), and ② identifying the sample from IDs of other source domains (hybrid scheme). Since the large domain discrepancy is much easier for the model to recognize, the back-propagated gradients from ② will focus more on the domain-sensitive parts of intermediate features than ① does. We can locate the domain-sensitive subset by comparing the gradients from both training schemes for better subsequent optimization. As shown in Fig. 1, due to the extremely different contrast between domain1 and domain2, the model can distinguish pedestrians in domain1 from domain2 only based on the contrast. However, the model learns the contrast feature may not be useful in other domains. This is harmful to the generalization ability of the model.

Our LEDF consists of three steps: (1) Constructing domain-specific & hybrid memory (2) Locating domain-sensitive features (3) Optimizing the generalized model. In step (1), we adopt the commonly used memory bank [42] to compute the classification losses. The "domain-specific memory" stores the class centroids of each ID for each domain, enabling us to compute the classification loss. The "hybrid memory" is the concatenation of all "domain-specific memories", which can be utilized to discriminate input samples under the training scheme ②. In step (2), we locate the domain-sensitive part of features by comparing gradients from two training schemes. Since the gradient from scheme ② will focus more on the domain-sensitive subsets of features (i.e., higher response), which is different from ①, we can compare the gradient of ① with the gradient of ② for better localization of these subsets. The located subsets will be muted in step (3) to train a generalized re-ID model.

Our main contributions are three-fold:

- We propose a novel domain generalization method for person re-ID by locating and eliminating the domain-sensitive subsets of features. After removing these features, the model can focus more on the generalized information in datasets and achieve better generalization.
- We design a novel strategy to locate domain-sensitive subsets of features by comparing the gradients from two different training schemes. As large domain discrepancy is easier for the model to recognize, the back-propagated gradients from the "hybrid scheme" will focus more on the domain-sensitive subsets of intermediate features than the "domain-specific scheme" does, enabling us to locate them in each back-propagation.
- Extensive experiments on four large-scale benchmarks verify the effectiveness of our method in multi-source domain generalization.

2 Related Work

2.1 Person Re-ID

With the rapid development of deep learning, person re-ID has made great progress in recent years [33, 35, 37, 32, 2, 6, 31, 41]. However, the performance of these methods in unseen domain testing is not satisfactory. Unsupervised Domain Adaptation (UDA) uses unlabelled target domain data to improve the model's performance in the unseen domains. ECN [42] pays more attention to the intra-domain variations of the target domain. MCD [26] uses adversarial learning to maximize the difference between the two different classifiers to detect the samples in the target domain far from the source domain. Then the feature generator is used to generate features similar to the source domain. AutoDIAL [1] proposes the domain alignment layer to automatically align the feature representation of different domains. MMT [4] is trained by two network mutual teaching methods to generate reliable pseudo labels, which solves the problem of pseudo labels noise in the target domain. ACT [32] uses two asymmetric models for collaborative training. The training samples of one model are as diverse as possible, and the training of the other model is as pure as possible. The two models give reliable samples to each other to avoid label noise.

2.2 Domain Generalization

Most DG methods are based on data augmentation, domain-invariant features learning, and gradient calibration. Data augmentation means generating data different from the source domain, learning more unprecedented features, and preventing over-fitting of the source domain. Domain-invariant features learning means learning more identify-relevant features. Gradient Calibration means using the gradient to design appropriate learning strategies to improve the generalization ability of the model.

Data Augmentation. L2A-OT [44] uses Optimal Transport (OT) to make the distribution between the generated image and the source domain image very different. MixStyle [45] combines the style features at the bottom layer of the network to generate new style features and enrich the diversity of training data. PDEN [16] simulates the unseen domain by constantly changing the brightness and geometry of the data on the source domain. FSDR [10] augments the image in frequency space. It keeps the domain-invariant frequency components as much as possible and randomizes the domain-variant frequency components in the frequency space. SFA [17] uses Gaussian noise to interfere with feature embedding in the training process, which improves the performance of the classifier in the unseen domain.

Domain-Invariant Features Learning. DICA [22] minimizes dissimilarity across domains to learn domain-invariant features. MMD-AAE [15] learns domain-invariant features through adversarial autoencoders. SNR [13] uses Instance Normalization to eliminate style features and restitutes the features to ensure that effective information is not filtered. IBN [23] points out that the low-level feature representation reflects more texture information, while the high-level feature representation reflects more semantic information. IBN uses the advantages of Instance Normalization and batch normalization to improve the generalization of models. MatchDG [20] makes use of the causal influence to provide an object-conditional objective to highlight the advantage of learning domain-invariant features. FAR [12] aligns the features of different domains by adjusting the moment modulation of feature distribution, then extracts useful features from the remaining information and uses them to compensate for the aligned features.

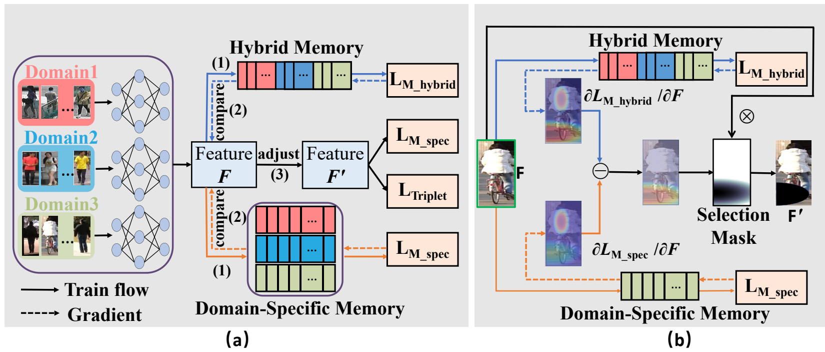

Fig. 2. Illustration of our proposed LEDF. (a) The overall framework of LEDF. (b) The detailed process of feature adjustment.

Gradient Calibration. Some methods solve DG problems through meta learning [14, 38]. MLDG [14] divides the data from the source domains into a meta-training set and meta-testing set to simulate the unseen domain. It lays a foundation for the later meta-DG methods. M^{3}L [38] uses the meta-test phase of meta-learning to simulate the unseen domain and uses metaBN to increase the diversity of meta-test features. Some methods use the gradient to further propose novel strategies [27, 21, 11]. Fish [27] points out that the learning ability of general features can be improved only if the gradient descent directions on the two domains are consistent. They improve the generalization ability by maximizing the inner product of the gradient descent directions of different domains. Gradient surgery [21] points out that intra-domain gradients often show higher similarity than inter-domain gradients. Consistency constraints on inter-domain gradients can encourage the learning of distinctive features common to all domains and improve the generalization performance across domains. RSC [11] points out that the model will over-rely on easy features, and improve the generalization ability of the model by filtering out features with large gradients.

3 Method

Problem Definition. Suppose we have N_S labeled source domains $\mathcal{D}_\mathcal{S} = \{\mathcal{D}_1, ..., D_{N_S}\}$, where $\mathcal{D}_i = \{\mathcal{X}_i, \mathcal{Y}_i\}$ is the i-th domain with training images $\mathcal{X}_i$ and their corresponding ID labels $\mathcal{Y}_i$. The goal of multi-domain generalized re-ID is optimizing re-ID model that can perform well on another unseen target domain $\mathcal{T}$ without using $\mathcal{T}$'s training images.

3.1 Overview

Our overall training framework is shown in Fig. 2-(a), which contains three steps. (1) designing domain-specific & hybrid training schemes (2) locating domain-sensitive features (3) generalized optimization. In step (1), we design two different training schemes for the same inputs and adopt memory banks to compute different losses. The first type of scheme (domain-specific scheme, also noted as scheme ①) aims at discriminating the inputs from their corresponding domain, while the second (hybrid scheme, also noted as scheme ②) tries to classify the given samples from IDs of all source domains. Since domain shift is much easier for the model to recognize due to the change of illumination and viewpoint [27], the back-propagated gradient from the hybrid scheme will focus more on the domain-specific subsets of features. Therefore, in step (2), we compare the back-propagated gradients of the two schemes to locate the domain-sensitive subsets. In step (3), we mute these subsets during the training process to improve the generalization of re-ID models. Fig. 2-(b) shows the process of feature adjustment in detail. Next, we will introduce our method in detail.

3.2 Domain-Specific & Hybrid Scheme

In this section, we introduce the two training schemes used in our method. The two training schemes are "domain-specific scheme" and "hybrid scheme". Given a batch of samples $\{\mathcal{X}_i, \mathcal{Y}_i\}$ with N_b images $\mathcal{X}_i = \{x_{i,1}, ..., x_{i,N_b}\}$ and labels $\mathcal{Y}_i = \{y_{i,1}, ..., y_{i,N_b}\}$ of domain i, the former discriminates them from IDs of their corresponding domain while the latter classifies them in IDs of all source domains. Generally speaking, we can simply adopt the fully-connected layer to compute the classification loss for these two schemes. However, as discussed in [38], re-ID is an open-set problem [24] where each domain has completely different IDs. We thus use the parametric classifier like memory bank [42] for optimization.

Memory Initialization. In our method, we construct N_S memory banks for each domain's domain-specific training. The memory bank for domain i is denoted as $\mathcal{M}_i \in \mathbb{R}^{N_i \times d}$, which is a $N_i \times d$ matrix and stores the L2-normed class centroids in each row of it. N_i is the number of IDs in the domain i while d is the dimension of features. We use imagenet-pretrained ResNet-50 model $f(\cdot)$ to extract features and class centroids for each domain. These class centroids are then L2-normalized to initialize the memories. Moreover, we concatenate all memories to form the hybrid memory $\mathcal{M}$ for our "hybrid scheme".

Loss Functions. We conduct our domain-specific and hybrid training schemes with all memories. Specifically, for a sample $\{x_{i,j}, y_{i,j}\}$ in the given batch $\{\mathcal{X}_i, \mathcal{Y}_i\}$, we compute domain-specific classification loss with its corresponding memory $\mathcal{M}_i$, formulated as:

$$L_{M\_spec} = -\log \frac{\exp(\mathcal{M}_i[y_{i,j}]^{\mathrm{T}} f(x_{i,j})/\tau)}{\sum_{k=1}^{N_i} \exp(\mathcal{M}_i[k]^{\mathrm{T}} f(x_{i,j})/\tau)} \ , \tag{1}$$

where $x_{i,j}$ is the j-th sample in the batch. $f(x_{i,j})$ denotes the L2-normalized intermediate feature of input image $x_{i,j}$. $\mathcal{M}_i[y_{i,j}]$ is the $y_{i,j}$-th class centroid in memory $\mathcal{M}_i$. τ is the temperature factor that controls the scale of distribution. N_i denotes the total number of IDs in the i-th domain. By using L_{M_spec} for domain-specific training, we enforce the model to classify the given sample into its own class as traditional re-ID methods do.

The motivation for designing the hybrid scheme is to discriminate the given sample from IDs of all source domains. Therefore, we concatenate all memory banks to form the hybrid memory $\mathcal{M} \in \mathbb{R}^{N \times d}$. $N = \sum_{i=1}^{N_S} N_i$ is the number of IDs in all domains. The training loss for the hybrid scheme is defined based on the hybrid memory:

$$L_{M\_hybrid} = -\log \frac{\exp(\mathcal{M}[y_{i,j}]^{\mathrm{T}} f(x_{i,j})/\tau)}{\sum_{k=1}^{N} \exp(\mathcal{M}[k]^{\mathrm{T}} f(x_{i,j})/\tau)} \ . \tag{2}$$

When discriminating the given sample from all centroids of source domains, it is usually easier for the model to focus on the domain-sensitive subsets of features. That is because the images in different domains may have conspicuous differences (*e.g.*, illumination and change of colors [34]), which is easier for the model to recognize than the fine-grained ID-related information [27]. Therefore, the back-propagated gradients from the two training schemes will be slightly different. We thus design a novel algorithm to locate this domain-sensitive information in the intermediate features by comparing gradients of two training schemes.

3.3 Locating Domain-Sensitive Features

We adopt the gradients from two training schemes to find the domain-sensitive subsets of features for subsequent generalization training. In detail, after computing the L_{M_spec} and L_{M_hybrid} for all samples in the training batch, we compute the derivative of these two losses respected to the intermediate features $\mathbf{F}$

$$g_{spec} = \partial L_{M\_spec} / \partial \mathbf{F}, \quad g_{hybrid} = \partial L_{M\_hybrid} / \partial \mathbf{F} \ , \tag{3}$$

where g_{spec} and g_{hybrid} are temporary back-propagated gradients of L_{M_spec} and L_{M_hybrid} for the training batch. As the hybrid training scheme focuses more on the domain-sensitive subsets of features, the gradient in these subsets will be higher than that gradient of the domain-specific scheme, while keeping the basic focus on generalized subsets to maintain the basic discrimination [11]. We thus compare g_{spec} with g_{hybrid} to obtain a response map and highlight these subsets:

$$\Delta rank(g_{\mathbf{F}}) = rank(g_{hybrid}) - rank(g_{spec}) \ , \tag{4}$$

where $rank$ means calculating the ranking value and $\Delta rank(g_{\mathbf{F}})$ is the response map. The subsets with higher values in $\Delta rank(g_{\mathbf{F}})$ indicate the higher possibility of being domain-sensitive. Based on the response map $\Delta rank(g_{\mathbf{F}})$, we find domain-sensitive subsets of features by locating the regions with higher response values. In detail, we sort all the values in $\Delta rank(g_{\mathbf{F}})$ in descent order and set a threshold q based on the top $r\%$ highest values of $\Delta rank(g_{\mathbf{F}})$. $r\%$ is the dropping rate. Subset features will be considered as the domain-sensitive parts if their corresponding response value in $\Delta rank(g_{\mathbf{F}})$ is greater than q. To formulate the selection process, we define a selection mask $\mathbf{m}$ that has the same size as intermediate feature $\mathbf{F}$. Each element in $\mathbf{m}$ can be computed with the following rule:

$$\mathbf{m}_o = \begin{cases} 0, & \text{if} \quad \Delta rank(g_{\mathbf{F},o}) \geq q \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

where $\mathbf{m}_o$ and $rank(g_{\mathbf{F},o})$ are the o-th element in $\mathbf{m}$ and $rank(g_{\mathbf{F}})$, respectively. q is the threshold for selection. The obtained mask is then used to filter out domain-sensitive subsets in $\mathbf{F}$ through element-wise product:

$$\mathbf{F}' = \mathbf{F} \odot \mathbf{m} \ , \tag{6}$$

where $\mathbf{F}'$ is the intermediate features that have muted the domain-sensitive subsets.

3.4 Generalized Optimization

The obtained $\mathbf{F}'$ is then sent to global average pooling and L2-norm layers through forward propagation to obtain adjusted features $\mathbf{f}^{'} \in \mathbb{R}^{N_b \times d}$. We use $\mathbf{f}^{'}$ to optimize the re-ID model with the following two losses:

$$L_{ce}(\mathbf{f}^{'}, \mathcal{Y}_i) = -\frac{1}{N_b} \sum_{j=1}^{N_b} \log \frac{\exp(\mathcal{M}_i[y_{i,j}]^{\mathrm{T}} \mathbf{f'}_j / \tau)}{\sum_{k=1}^{N_i} \exp(\mathcal{M}_i[k]^{\mathrm{T}} \mathbf{f'}_j / \tau)} \ , \tag{7}$$

$$L_{tri}(\mathbf{f}^{'}) = \frac{1}{N_b} \sum_{j=1}^{N_b} \Big[||\mathbf{f}^{'}_{+,j} - \mathbf{f}^{'}_j||_2 - ||\mathbf{f}^{'}_{-,j} - \mathbf{f}^{'}_j||_2 + m \Big]_+ \ , \tag{8}$$

where $\mathbf{f}^{'}_j$ is the j-th feature in $\mathbf{f}'$. $\mathbf{f}^{'}_{+,j}$ and $\mathbf{f}^{'}_{-,j}$ are hard positive and negative samples within the batch. m is the margin for triplet loss. The final loss for optimizing re-ID model is defined as:

$$\mathcal{L}_{all} = \mathcal{L}_{ce}(\mathbf{f}', \mathcal{Y}_i) + \mathcal{L}_{tri}(\mathbf{f}') \ . \tag{9}$$

At the end of optimization, we update the memory of domain i to prepare for the training with next batch of samples. Specifically, the corresponding class centroids in the i-th memory bank will be updated with the exponential moving average strategy, which is commonly used in memory-based domain generalization methods [38]:

$$\mathcal{M}_i[z] \leftarrow \mu \cdot \mathcal{M}_i[z] + (1 - \mu) \cdot \frac{1}{|\mathcal{B}_z|} \sum_{x_{i,j} \in \mathcal{B}_z} f(x_{i,j}) \ , \tag{10}$$

where $\mathcal{B}_z$ denotes the samples belonging to the z-th ID and $|\mathcal{B}_z|$ denotes the number of samples for the z-th ID in the current mini-batch. $\mu \in [0, 1]$ controls the updating rate. The previous three steps (step (1)-(3)) are iterated for several epochs to improve the model's generalization. The overall training process has been demonstrated in Alg. 1.

Algorithm 1: Procedure of Our Algorithm.

Inputs: N_S Labeled source domains $\mathcal{D}_\mathcal{S} = \{\mathcal{D}_1, ..., D_{N_S}\}$, learning rate β, batch size N_b, training epochs *epoch*.
Outputs: Generalized re-ID model.
1: Initialize all memories.
2: **for** s in *epoch* **do**
3: **for** i in N_S **do**
4: Sample a mini-batch $\{\mathcal{X}_i, \mathcal{Y}_i\}$ with N_b images and labels from domain i;
5: Compute temporary gradient g_{spec} and g_{hybrid} with Eq. 3;
6: Locate domain-sensitive subsets with Eq. 4;
7: Generate $\mathbf{F}'$ with Eq. 6;
8: Optimize re-ID model with Eq. 9;
9: Update memory with Eq. 10;
10: **end for**
11: **end for**
12: Return generalized re-ID model.

Further Discussion. Our method is inspired by the gradient-based generalization algorithm RSC [11]. However, our approach is fundamentally different from it. Specifically, our method considers the interaction among source domains by simultaneously using domain-specific and hybrid memory, while RSC does not have such interaction. The interaction of source domains enables us to define a more specific meaning of domain-shift, and make the assumption that when compared with the domain-specific scheme, the subsets of features with higher predictability in the hybrid scheme have a higher possibility of being domain-sensitive. The domain-sensitive subsets can be readily located when comparing the back-propagated gradients of the two schemes. However, RSC considers the most predictive subsets of features as the easiest part and mutes them during the optimization for generalized training. They do not explicitly consider the domain shift problem. Therefore, our method is different from RSC and we design a non-trivial solution for generalized re-ID learning.

4 Experiments

4.1 Experiment Settings

Datasets. We selected four large-scale person re-identification benchmarks for experiments, *i.e.*, Market-1501 [39], CUHK03 [19], CUHK02 [18], and MSMT-17 [29]. Market-1501 was collected by 6 cameras, of which $32,668$ labeled data composed of 1501 IDs. CUHK03 and CUHK02 have $1,467$ and $1,816$ IDs, $28,193$ and $7,264$ pedestrian images respectively. And they have five pairs of different outdoor cameras. MSMT-17 is an extra-large pedestrian dataset, including $126,441$ pedestrian images collected by 15 cameras, with a total of $4,101$ IDs. Note that, we do not take DukeMTMC-reID [40, 25] into our evaluation since it has been taken down by its creators.

Evaluation Metrics. We evaluated the performance based on the cumulative matching characteristics (CMC) at Rank-1 and mean average precision (mAP).
Implementation Details. We conduct our experiments on two commonly used backbones, *i.e.*, ResNet-50 [7] and IBN-Net50 [23]. We apply our method in the intermediate feature maps. We set the total training epochs to 60. We choose Adam as the optimizer and the initial learning rate is set to 3.5×10^{-5}. We multiply the learning rate by 0.1 at the 30-th and 50-th training epoch. For the loss function, we set the margin of triple loss m=0.3. The temperature factor of memory-based identification loss τ=0.05 and updating rate μ in memory update is set to 0.2. We set the training batch-size N_b to 64. All images are resized to 256×128, followed by random flipping and random clipping for data augmentation. We adopt the re-ID model trained with all source domains and vanilla memory-based identification loss and triplet loss [8] as the baseline.

4.2 Comparison with State-of-the-arts Methods

We compare our method with state-of-the-art methods and report the results in Tab. 1. The included methods are SNR [13], RSC [11], and M^{3}L [38]. We evaluate the generalization of the trained re-ID model with the "leave-one protocol", *i.e.*, testing the model on one of the four benchmarks and using other datasets as source domains. As shown in the table, our method outperforms other methods by a large margin in both mAP and rank-1 scores. Specifically, the experiments with ResNet-50 achieve 53.8%, 50.6%, 89.6%, and 15.7% mAP scores when using Market1501, CUHK03, CUHK02, and MSMT17 as the testing set. These results outperform M^{3}L [38] by 3.9%, 5.5%, 2.7%, and 4.6% on the previous four generalization tasks, respectively. Moreover, when applying our method to IBN-Net50, our method also achieves the best re-ID accuracies. The mAP scores of the re-ID model are 58.6%, 53.8%, 90.0%, and 18.9%, which is higher than that of M^{3}L by 6.9%, 3.6%, 1.8%, and 4.9% when being evaluated on Market1501, CUHK03, CUHK02, and MSMT17. Similar results can also be found in another state-of-the-art method RSC [11], which is originally designed for the classification task. Based on these results, we claim that our method is effective in handling the generalized re-ID model training problem.

4.3 Ablation Studies

To further explore the effectiveness of each component in our method, we design three ablation experiments. In these experiments, we aim to: (1) show the necessity of eliminating domain-sensitive subsets of features during the optimization; (2) prove the necessity of locating domain-sensitive subsets by subtracting $rank(g_{spec})$ from $rank(g_{hybrid})$; (3) prove the effectiveness of eliminating domain-sensitive subsets with our method.
The Necessity of Eliminating Domain-Sensitive Subsets. We evaluate the re-ID model that does not use any strategy to eliminate domain-sensitive subsets during the optimization and report the results in Tab. 2. In ResNet-50, the model trained with vanilla multi-domain training (baseline) achieves 46.1%

Table 1. Comparison with State-of-the-arts. M: Market-1501, CUHK02: C2, CUHK03: C3, MS: MSMT-17.

Methods	Sources	Market		Sources	CUHK03	
		mAP	rank-1		mAP	rank-1
OSNet [43]	MS+C2+C3	41.7	65.5	MS+M+C2	39.4	41.1
SNR [13]		44.2	70.1		41.2	45.5
RSC [11]		50.8	76.7		46.5	51.5
M^{3}L [38]		49.9	75.7		45.1	50.5
Ours		53.8	79.0		50.6	56.0
M^{3}L (IBN-Net50)		51.7	78.0		50.2	57.5
Ours (IBN-Net50)		**58.6**	**81.8**		**53.8**	**59.8**

Methods	Sources	CUHK02		Sources	MSMT	
		mAP	rank-1		mAP	rank-1
OSNet [43]	MS+C2+C3	75.5	76.1	MS+M+C2	10.2	26.7
SNR [13]		78.5	79.5		14.4	37.3
RSC [11]		88.2	87.7		13.1	33.1
M^{3}L [38]		86.9	87.4		11.1	28.8
Ours		89.6	**90.6**		15.7	39.5
M^{3}L (IBN-Net50)		88.2	87.6		14.0	34.2
Ours (IBN-Net50)		**90.0**	89.1		**18.9**	**44.2**

Table 2. Ablation studies on Locating strategy. H-S: computing response map by subtracting $rank(g_{spec})$ from $rank(g_{hybrid})$. S-H: computing response map by subtracting $rank(g_{hybrid})$ from $rank(g_{spec})$. Abs: Using the absolute values of the difference between $rank(g_{hybrid})$ and $rank(g_{spec})$ to compute response map. ResNet: ResNet-50. IBN-Net: IBN-Net50.

Backbone	Strategy			MS+C2+ C3→M		M+C2+ C3→MS		Backbone	MS+C2+ C3→M		M+C2+ C3→MS	
	Abs	S-H	H-S	mAP	rank-1	mAP	rank-1		mAP	rank-1	mAP	rank-1
ResNet	×	×	×	46.1	72.3	9.5	25.0	IBN-Net	50.5	75.7	12.6	32.3
	✓	×	×	53.0	78.4	14.7	38.1		57.8	80.9	17.9	42.7
	×	✓	×	52.9	78.1	14.4	37.6		57.2	79.5	17.2	41.3
	×	×	✓	**53.8**	**79.0**	**15.7**	**39.5**		**58.6**	**81.8**	**18.9**	**44.2**

and 9.5% mAP scores on "MS+C2+C3→M" and "M+C2+C3→MS" tasks, respectively. The results are lower than our method in these two tasks. Similarly, in IBN-Net50, our experiment also achieved the same conclusion. This experiment indicates that eliminating the domain-sensitive subsets of features during the optimization is necessary.

The Necessity of Subtracting $rank(g_{spec})$ from $rank(g_{hybrid})$. In Eq. 4, we generate the response map by subtracting $rank(g_{spec})$ from $rank(g_{hybrid})$. Intuitively, there are also other strategies to highlight the domain-sensitive subsets of features, like using the absolute value of the difference between $rank(g_{spec})$ and $rank(g_{hybrid})$ (*i.e.*, $\Delta rank(g_{\mathbf{F}}) = |rank(g_{hybrid}) - rank(g_{spec})|$). To explore the best form of highlighting these domain-sensitive subsets of features, we conduct experiments by using different strategies to compute response map $\Delta rank(g_{\mathbf{F}})$

Table 3. Ablation studies of dropping strategies.

Backbone	Method	MS+C2+C3 → M		M+C2+C3 → MS	
		mAP	rank-1	mAP	rank-1
ResNet-50	Random	48.5	73.9	9.3	25.2
	RSC	50.8	76.7	13.1	33.1
	Ours	**53.8**	**79.0**	**15.7**	**39.5**

in Tab. 2. As shown in the table, the strategy define in Eq. 4 achieves the best performance when compared with other strategies. This suggests that subtracting $rank(g_{spec})$ from $rank(g_{hybrid})$ is a more plausible way of highlighting the domain-sensitive subsets during the optimization.

The Necessity of Using Our Dropping Strategy. We evaluate the re-ID model by randomly dropping features. The random dropout strategy adopts the same settings as our method, such as the same feature dropping rate, etc. As shown in Tab. 3, The random dropout strategy achieves 48.5% and 9.3% mAP scores on "MS+C2+C3→M" and "M+C2+C3→MS" tasks, respectively. The results are lower than RSC and our method, and our method achieves the best. This shows that our method is most effective in eliminating domain-sensitive features.

4.4 Sensitivity Analysis

Feature Dropping Rate. We explore the influence of using different dropping rates during the optimization. The higher dropping rate r may inevitably drop some decisive features while lower r may not be effective enough to improve the generalization of models. In Tab. 4, we vary r from 10% to 66.7% on "MS+C2+C3→M" and "MS+C2+M→C3" to find how the dropping rate will influence the final results. From Tab. 4, we find that: (1) When the dropping rate r increases from 10% to 50%, the re-ID accuracies in generalization tasks also improved. This is caused by the elimination of domain-sensitive subsets of features. (2) When continuously increasing the dropping rate r from 50% to 66.7%, the re-ID accuracies decreases. The results indicate that we should not set the dropping rate to a very high value. Generally speaking, setting r to a value less than 50% would be sufficient for improving the model's generalization.

Number of Source Domains. We also conduct experiments using fewer source domains to check the effectiveness of our method. In detail, we alternately use two of the three source domains for training and evaluate the trained model on the testing set. The results are shown in Tab. 5. We report the re-ID accuracies of re-ID models on "MS+C3→M", "MS+M→C3", and "M+C3→MS" tasks. For simplicity, in addition to baseline, we also compare it with RSC, which has the best average performance in previous experiments among many state-of-the-art methods. As shown in the table, we achieve 46.5%, 36.8%, and 13% mAP scores on "MS+C3→M", "MS+M→C3", and "M+C3→MS" tasks, respectively. These results outperform RSC [11] and the vanilla generalized re-ID training method,

Table 4. Sensitivity analysis of dropping rate.

Rate	MS+C2+C3 → M		MS+M+C2 → C3		Average	
	mAP	rank-1	mAP	rank-1	mAP	rank-1
10.0%	52.3	77.9	48.7	53.0	50.5	65.45
20.0%	53.0	78.4	48.6	52.5	50.8	65.45
25.0%	53.6	78.9	48.8	55.5	51.2	67.20
33.3%	53.4	78.8	**50.8**	54.0	52.1	66.40
50.0%	**53.8**	**79.0**	50.6	**56.0**	**52.2**	**67.50**
66.7%	53.3	78.1	46.5	54.0	49.9	66.05

Table 5. Sensitivity analysis of the number of source domains.

Backbone	Method	MS+C3 → M		MS+M → C3		M+C3 → MS	
		mAP	rank-1	mAP	rank-1	mAP	rank-1
ResNet-50	Baseline	38.5	64.6	29.9	34.5	6.8	18.8
	RSC	44.0	68.6	35.7	42.3	9.9	26.0
	Ours	**46.5**	**73.0**	**36.8**	**43.5**	**13.0**	**33.4**

Table 6. Sensitivity analysis of Channel-wise LEDF verse Spatial+Channel LEDF.

Backbone	Method	MS+C2+C3 → M		MS+M+C2 → C3	
		mAP	rank-1	mAP	rank-1
ResNet-50	Channel	50.7	77.6	46.5	54.0
	Channel+Spatial	**53.8**	**79.0**	**50.6**	**56.0**

which proves the effectiveness of our method on generalization tasks with fewer source training domains.

Channel-Wise and Spatial-Wise Dropping. We also design some experiments to find the optimal strategy for dropping intermediate features. Intuitively, there are two types of dropping strategies. The first is channel-wise dropping, which means using the global average pooling to gradient tensor along the spatial dimension and carrying out feature dropping at the channel level. And the second is spatial-wise dropping. Because ResNet uses global average pooling at the end, the values of applying global average pooling to gradient tensor along the channel dimension are all the same. Therefore, we use the same spatial-wise dropping as RSC [11], which is transformed on the basis of channel-wise dropping. As shown in Tab. 6, the combination of channel-wise dropping and spatial-wise dropping performs best (we use 50% of channel-wise dropping and spatial-wise dropping respectively in the training stage).

What Stage to Add Our Method? It naturally raises the question that how about adding our method to other intermediate features. To answer this question, we take the ResNet-50 model as an example and insert our method in different stages of the model to check its effectiveness. Generally speaking, ResNet-50 has four blocks, and we can insert our method at the beginning or end of each block. Therefore, we apply our method in the position depicted in Fig. 3 and report the

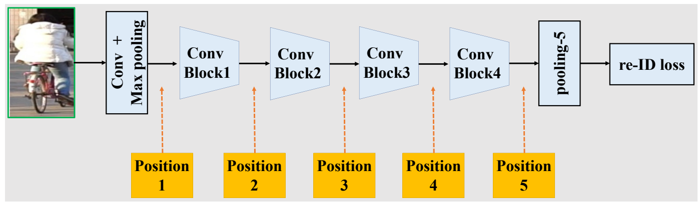

Fig. 3. Our LEDF module can be inserted into the position of ResNet-50.

Table 7. Sensitivity analysis of LEDF inserts position.

Position	MS+C2+C3 $\rightarrow$ M		MS+M+C2 $\rightarrow$ C3		Average	
	mAP	rank-1	mAP	rank-1	mAP	rank-1
Position1	**53.8**	**79.0**	**50.6**	**56.0**	**52.2**	**67.50**
Position2	51.5	77.0	45.7	47.0	48.6	62.00
Position3	46.8	73.8	43.6	49.5	45.2	61.65
Position4	42.1	69.5	40.5	48.5	41.3	59.00
Position5	36.4	64.0	37.6	46.0	37.0	55.00

results in Tab. 7. We conclude that our method achieves the best performance when being deployed in the shallow layer while performing poorly when being deployed in deeper layers. We conjecture that style information is stored in the shallow layers of deep networks while deep layers are responsible for learning semantic information. As our method aims at eliminating the domain-shift of features, it is better to deploy our method in the shallow layers of deep neural networks.

5 Conclusion

In this paper, we propose a novel domain generalization method to enhance the generalization of the model in the unseen domain by locating and eliminating domain-sensitive features (LEDF). In addition, we introduced domain-specific memory and hybrid memory, which respectively represent two different training schemes. The former only has the information of its own domain, and the latter contains the information of all source domains. LEDF locates and eliminates the domain-sensitive features by comparing the gradient between the two schemes. A large number of experiments prove that our method LEDF can effectively improve the generalization ability of the model in person re-identification.

Acknowledgement This work is supported by the National Nature Science Foundation of China (No. 61876159, 62276221), the Natural Science Foundation of Fujian Province of China (No. 2022J01002), the Science and Technology Plan Project of Xiamen (No. 3502Z20221025).

References

1. Carlucci, F.M., Porzi, L., Caputo, B., Ricci, E., Bulo, S.R.: Autodial: Automatic domain alignment layers. In: ICCV (2017)
2. Ding, C., Wang, K., Wang, P., Tao, D.: Multi-task learning with coarse priors for robust part-aware person re-identification. TPAMI (2020)
3. Fan, H., Zheng, L., Yan, C., Yang, Y.: Unsupervised person re-identification: Clustering and fine-tuning. TOMM (2018)
4. Ge, Y., Chen, D., Li, H.: Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In: ICLR (2020)
5. Ge, Y., Zhu, F., Chen, D., Zhao, R., et al.: Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In: NIPS (2020)
6. Gong, X., Yao, Z., Li, X., Fan, Y., Luo, B., Fan, J., Lao, B.: Lag-net: Multi-granularity network for person re-identification via local attention system. TMM (2021)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
8. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
9. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
10. Huang, J., Guan, D., Xiao, A., Lu, S.: Fsdr: Frequency space domain randomization for domain generalization. In: CVPR (2021)
11. Huang, Z., Wang, H., Xing, E.P., Huang, D.: Self-challenging improves cross-domain generalization. In: ECCV (2020)
12. Jin, X., Lan, C., Zeng, W., Chen, Z.: Feature alignment and restoration for domain generalization and adaptation. arXiv preprint arXiv:2006.12009 (2020)
13. Jin, X., Lan, C., Zeng, W., Chen, Z., Zhang, L.: Style normalization and restitution for generalizable person re-identification. In: CVPR (2020)
14. Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Learning to generalize: Meta-learning for domain generalization. In: AAAI (2018)
15. Li, H., Pan, S.J., Wang, S., Kot, A.C.: Domain generalization with adversarial feature learning. In: CVPR (2018)
16. Li, L., Gao, K., Cao, J., Huang, Z., Weng, Y., Mi, X., Yu, Z., Li, X., Xia, B.: Progressive domain expansion network for single domain generalization. In: CVPR (2021)
17. Li, P., Li, D., Li, W., Gong, S., Fu, Y., Hospedales, T.M.: A simple feature augmentation for domain generalization. In: ICCV (2021)
18. Li, W., Wang, X.: Locally aligned feature transforms across views. In: CVPR (2013)
19. Li, W., Zhao, R., Xiao, T., Wang, X.: Deepreid: Deep filter pairing neural network for person re-identification. In: CVPR (2014)
20. Mahajan, D., Tople, S., Sharma, A.: Domain generalization using causal matching. In: ICML (2021)
21. Mansilla, L., Echeveste, R., Milone, D.H., Ferrante, E.: Domain generalization via gradient surgery. In: ICCV (2021)
22. Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML (2013)
23. Pan, X., Luo, P., Shi, J., Tang, X.: Two at once: Enhancing learning and generalization capacities via ibn-net. In: ECCV (2018)
24. Panareda Busto, P., Gall, J.: Open set domain adaptation. In: ICCV (2017)

25. Ristani, E., Solera, F., Zou, R., Cucchiara, R., Tomasi, C.: Performance measures and a data set for multi-target, multi-camera tracking. In: ECCV Workshop (2016)
26. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: CVPR (2018)
27. Shi, Y., Seely, J., Torr, P.H., Siddharth, N., Hannun, A., Usunier, N., Synnaeve, G.: Gradient matching for domain generalization. In: ICLR (2022)
28. Sun, Y., Zheng, L., Yang, Y., Tian, Q., Wang, S.: Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In: ECCV (2018)
29. Wei, L., Zhang, S., Gao, W., Tian, Q.: Person transfer gan to bridge domain gap for person re-identification. In: CVPR (2018)
30. Xiang, W., Huang, J., Hua, X.S., Zhang, L.: Part-aware attention network for person re-identification. In: ACCV (2020)
31. Yan, C., Pang, G., Bai, X., Liu, C., Xin, N., Gu, L., Zhou, J.: Beyond triplet loss: person re-identification with fine-grained difference-aware pairwise loss. TMM (2021)
32. Yang, F., Li, K., Zhong, Z., Luo, Z., Sun, X., Cheng, H., Guo, X., Huang, F., Ji, R., Li, S.: Asymmetric co-teaching for unsupervised cross-domain person re-identification. In: AAAI (2020)
33. Yang, F., Zhong, Z., Liu, H., Wang, Z., Luo, Z., Li, S., Sebe, N., Satoh, S.: Learning to attack real-world models for person re-identification via virtual-guided meta-learning. In: AAAI (2021)
34. Yang, F., Zhong, Z., Luo, Z., Cai, Y., Lin, Y., Li, S., Sebe, N.: Joint noise-tolerant learning and meta camera shift adaptation for unsupervised person re-identification. In: CVPR (2021)
35. Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., Hoi, S.C.: Deep learning for person re-identification: A survey and outlook. TPAMI (2021)
36. Yu, H.X., Zheng, W.S., Wu, A., Guo, X., Gong, S., Lai, J.H.: Unsupervised person re-identification by soft multilabel learning. In: CVPR (2019)
37. Zhang, Z., Lan, C., Zeng, W., Jin, X., Chen, Z.: Relation-aware global attention for person re-identification. In: CVPR (2020)
38. Zhao, Y., Zhong, Z., Yang, F., Luo, Z., Lin, Y., Li, S., Sebe, N.: Learning to generalize unseen domains via memory-based multi-source meta-learning for person re-identification. In: CVPR (2021)
39. Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., Tian, Q.: Scalable person re-identification: A benchmark. In: ICCV (2015)
40. Zheng, Z., Zheng, L., Yang, Y.: Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In: ICCV (2017)
41. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: CVPR (2017)
42. Zhong, Z., Zheng, L., Luo, Z., Li, S., Yang, Y.: Invariance matters: Exemplar memory for domain adaptive person re-identification. In: CVPR (2019)
43. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning generalisable omni-scale representations for person re-identification. TPAMI (2021)
44. Zhou, K., Yang, Y., Hospedales, T., Xiang, T.: Learning to generate novel domains for domain generalization. In: ECCV (2020)
45. Zhou, K., Yang, Y., Qiao, Y., Xiang, T.: Domain generalization with mixstyle. In: ICLR (2021)
46. Zhu, Z., Jiang, X., Zheng, F., Guo, X., Huang, F., Sun, X., Zheng, W.: Aware loss with angular regularization for person re-identification. In: AAAI (2020)