2-2023

# Learning comprehensive global features in person re-identification: Ensuring discriminativeness of more local regions

Jiali XIA

Jianqiang HUANG

Shibao ZHENG

Qin ZHOU

Bernt SCHIELE

*See next page for additional authors*

# Learning Comprehensive Global Features in Person Re-Identification: Ensuring Discriminativeness of More Local Regions

Jiali Xi[a], Jianqiang Huang[b], Shibao Zheng[a,*], Qin Zhou[c], Bernt Schiele Fellow, IEEE,[d], Xian-Sheng Hua Fellow, IEEE,[b], Qianru Sun[e]

[a]*Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, 200240, Shanghai, China*
[b]*Damo Academy, Alibaba Group, 310024, Hangzhou, China*
[c]*Institute of Medical Robotics, Shanghai Jiao Tong University, 200240, Shanghai, China*
[d]*Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken 66123, Germany*
[e]*School of Computing and Information Systems, Singapore Management University, 178902, Singapore*

**Abstract**

Person re-identification (Re-ID) aims to retrieve person images from a large gallery given a query image of a person of interest. Global information and fine-grained local features are both essential for the representation. However, global embedding learned by naive classification model tends to be trapped in the most discriminative local region, leading to poor evaluation performance. To address the issue, we propose a novel baseline network that learns strong global feature termed as Comprehensive Global Embedding (CGE), ensuring more local regions of global feature maps to be discriminative. In this work, two key modules are proposed including Non-parameterized Local Classifier (NLC) and Global Logits Revise (GLR). The NLC is designed to obtain a score vector of each local region on feature maps in a non-parametric manner. The GLR module directly revises the logits such that the subsequent cross entropy loss up-weights the loss assigned to samples with hard-to-learn local regions. The convergence

*Corresponding author

*Email addresses:* xijiali@sjtu.edu.cn (Jiali Xi), jianqiang.jqh@gmail.com (Jianqiang Huang), sbzh@sjtu.edu.cn (Shibao Zheng), sunnyzq1990@gmail.com (Qin Zhou), schiele@mpi-inf.mpg.de (Bernt Schiele Fellow, IEEE,), xiansheng.hxs@alibaba-inc.com (Xian-Sheng Hua Fellow, IEEE,), qianrusun@smu.edu.sg (Qianru Sun)

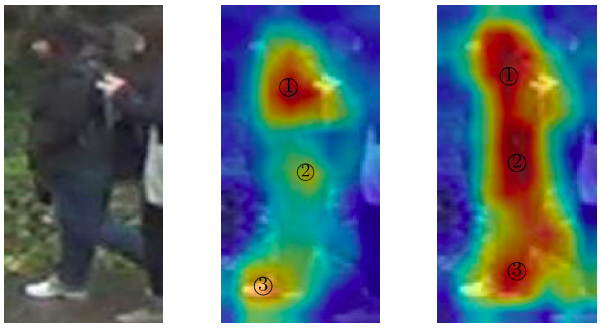
of the deep model indicates more local regions (the number of local regions is manually defined) on the feature maps of each sample are discriminative. We implement these two modules on two strong baseline methods including the BagTricks (BOT) [1] and AGW [2]. The network achieves 65.9% mAP, 85.1% rank1 on MSMT17, 86.4% mAP, 87.4% rank1 on CUHK03 labeled, 84.2% mAP, 85.9% rank1 on CUHK03 detected, and 92.2% mAP, 96.3% rank1 on Market-1501. The results show that the proposed baseline achieves a new state-of-the-art when using only global embedding during inference without any re-ranking technique.

*Keywords:* person re-identification, baseline, comprehensive

## 1. Introduction

Person re-identification (Re-ID) aims to associate images of the same person across non-overlapping camera views. It plays an important role in a number of practical scenarios of intelligent surveillance systems such as tracking and person activity analysis. Re-ID is extremely challenging and remains unsolved due to various reasons. First of all, under different camera views, a person's appearance often changes drastically due to variations on body posture, camera view, occlusion and lighting conditions. Second, in public places, many people often wear very similar clothes (e.g., white T-shirt in summer), leading to the discriminative clues existing in subtle region.

Early Re-ID works delicate to design hand-crafted features which are invariant to background, viewpoint as well as pose changes, and to conduct robust distance metric learning such as XQDA [3], KISSME [4]. Recent Re-ID algorithms deploy deep convolutional neural networks (CNNs) to learn deep features that are robust to those variations. Top performing methods often learn fine-grained embeddings from local image regions, such as spatial stripes [5, 6], semantic patches [7] and either of them with multiple scales [8] , particularly effective against partial appearance changes and background clutter [6]. Nevertheless, local features lack global information. Existing methods [8, 9] that




(a) Input (b) IDE (c) CGE

Figure 1: Saliency maps. (a) is the input image (b) is the IDE feature extracted by ResNet-50 [10], the local regions ②, ③ of which are less discriminative compared with region ①. (c) represents our proposed CGE which is comprehensive. The saliency maps are drawn via taking the maximum of feature maps along the channel dimension [9].

blend global and local clues together should be the optimal choice. A fly in the ointment is that naive global feature has the problem of being trapped in the most discriminative local region.

In this paper, we aim to tackle this problem by proposing a novel baseline network to learn comprehensive global embedding, ensuring that more local regions on feature maps are discriminative, as shown in Figure 1. How to realize this? We propose to use local discriminative ability to revise the global logits, thus manipulating the loss value. Specifically, less dicriminative regions (e.g. ②, ③ in Figure 1) will enlarge the loss value computed on the global embedding, thus increasing the gradient and assigning larger weights to these hard samples. By contrary, if all the pre-defined local regions are discriminative, the loss value based on global embedding will decrease and these samples will be assigned low importance. The convergence of the deep model indicates more regions on the feature maps of each sample are discriminative. We term the module that revises global logits as Global Logits Revise (GLR), and the module that obtains local discriminative ability as Non-parameterized Local Classifier (NLC).

The idea of making the local regions of the global feature discriminative via regularization is very similar to random erasing augmentation (REA) [11]. REA

randomly erases a certain part of an input image. The network has to focus on other regions to identify the person, which leads to more local areas that are discriminative in the global feature. Nevertheless, the impact of REA is random and in a global statistical sense. By contrast, the regularization effect of our method is controllable and specific to each training sample. The network actively adjusts the scale of its gradients to assign larger weights to the samples that violate the constraint.

From the view of re-weighting, our method is also similar to focal loss [12]. Focal loss makes a modification on the original cross entropy loss to make hard negative samples play a leading role in training. The differences between our method and focal loss lie in two folds. The focal loss relies on the confidence computed on the global feature to adjust the importance of samples, while our method makes the adjustment more fine-grained. Instead of classifying samples into easy and hard ones, CGE can divide them into the easiest, mid-easy, mid-hard and hardest. For example, if all the pre-defined local regions of an image are discriminative, it will be regarded as an easiest sample and assigned the lowest weight. On the contrary, if all the regions are mis-classified, the sample will be assigned the highest weight. Second, our method exhibits better performance in the Re-ID task according to vast experiments.

To sum up, this is the first work to utilize the discriminativeness of local regions as guidance to directly revise the global logits, ensuring the comprehensiveness of global features. Our contribution is thus three-fold. (i) We propose two novel modules named as Global Logits Revise (GLR) and Non-parameterized Local Classifier (NLC). The former module is novel while the latter is similar to memory [13]. The main difference lies in that CGE uses multiple memory banks to store the class-wise local features, aiming to measure the discriminative ability of each region on the global feature maps, which are then used as guidance in the GLR module to revise the global logits. [13] proposed to use a memory bank to store the instance representation vectors of unlabeled data in the area of domain adaptive Re-ID, aiming to introduce invariance learning. (ii) The proposed baseline is versatile and can be used to enhance various existing Re-ID

methods based on global features, e.g. BagTricks (BOT) [1] and AGW [2]. (iii) We conduct extensive experiments on three popular Re-ID benchmarks including Market-1501, CUHK03 and MSMT17. We achieve state-of-the-arts results on all these datasets with only global embedding during inference without any re-ranking techniques or auxiliary information such as semantic attributes or viewpoint information.

## 2. Related Work

According to [2], we discuss the feature learning strategies in Re-ID mainly from three distinct patterns: (1) Global (2) Local and (3) Auxiliary feature. After that, we will emphasize the main differences between the proposed CGE and existing methods that (1) apply random erasing augmentation (2) revise cross entropy loss and (3) introduce attention mechanism.

### *2.1. Global feature representation learning*

Global feature representation learning extracts a compact global embedding for each person image. IDE [14] learns global features in a simple identity classification framework. This method is widely used in the Re-ID community nowadays. [15] proposed to combine batch-hard triplet loss and classification loss based on global features to improve the Re-ID accuracy. BagTricks (BOT) [1] proposed a bag of tricks including warm-up , random erasing [11], label smoothing , BNNeck [1] *etc.*, to improve the Re-ID performance based on global features. AGW [2] designed a new baseline for Re-ID, containing non-local attention block [16], generalized-mean (GeM) pooling and weighted regularization triplet (WRT) loss. This method has achieved the state-of-the-arts performance under multiple Re-ID benchmarks via only global representation.

### *2.2. Local feature representation learning*

Local feature representation learning usually extracts local features either via roughly-divided stripes [6], pose estimation [17] or human parsing [7]. PCB [6] divided the feature maps before the global max pooling layer into several stripes

to produce local features. Based on PCB, MGN [8] proposed to integrate the discriminative information obtained from different sizes of partitioned stripes which are assumed to contain various granularity. Deep-Person [5] applied Long Short-Term Memory (LSTM) on local features to capture the context information about each identity. PL-Net [9] proposed to compute an individual identity classification loss on each detected human part, and to concatenate global and local features to be the final representation of the image. PDC [17] proposed to align the local features under different poses. ISP [7] proposed to cluster the pixel-level features to generate pseudo-labels and then re-train the network in a self-supervised manner. Recently, some methods introduced Graph Convolutional Network (GCN) to capture the local relation among parts, which has achieved high performances. For example, HLGAT [18] and PGCN [19] proposed to model both the inter-local and intra-local relation between local features.

### *2.3. Auxiliary feature representation learning*

Auxiliary feature representation learning usually makes use of other supervision including attributes, pseudo-labels or frame stamp information in addition to identity labels to tackle Re-ID. VA-reID [20] made use of viewpoint pseudo-labels in a soft manner. st-ReID [21] utilized the spatiotemporal information of the input images. [22] added attribute annotations on two large-scale benchmarks to learn more informative Re-ID embeddings. [23] leveraged the off-the-shelf dense pose detector to automatically detect "pose annotations". [24] made use of texture image as additional supervision generated by the model trained with synthesized person images.

### *2.4. Erasing-related methods*

[11] proposed Random Erasing Augmentation (REA) that randomly selects a rectangle region in an image and erases its pixels with random values or the mean values of ImageNet [25]. [26] introduced a Batch DropBlock that randomly drops the same region of all input feature maps in a batch to reinforce the feature

learning of the rest local regions. [27] proposed SpatialDropout that zeros out one(several) certain spatial region(s) on the feature maps. [28] proposed SCSN to suppress the salient features learned in the previous cascaded stage, and then to extract other potential salient features in the following stage, aiming to obtain different clues of pedestrians. The first three methods are all derived from dropout. The proposed method differs from them in its controllability and pertinence. SCSN [28] requires a specific fusion strategy, while our method directly derives the comprehensive person representation.

### *2.5. Modifying cross entropy loss*

[12] proposed focal loss to reshape the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples. Our method manipulates cross entropy loss in a more fine-grained manner. We divide the training samples into multiple hardness levels and assign corresponding loss values according to the discriminative ability of multiple local regions.

### *2.6. Attention-based methods*

Attention mechanism is widely applied in Re-ID to enhance pedestrian representations. HA-CNN [29] extracted spatial attention to be the weights on different body regions, channel-wise attention to catch the dependency, and hard attention to yield local stripes. MEMF [30] proposed multi-level attention and multi-layer feature fusion to enrich the global features. SDN [31] proposed to capture inter-image and intra-image correlations to solve the spatial dependency problem. Whether the Re-ID model introduces spatial, channel-wise, or other forms of attention, weights need to be calculated to strengthen the naive features in the inference stage. However, our method gets rid of such constraints, since the local region enhancement is conducted in the training of the backbone via the proposed GLR and NLC modules. A very recent work DAAF-BoT [32] proposed to implicitly embed the attention information into the feature maps without modifying the backbone. They utilized two branches guided by mask prediction loss and keypoint prediction loss respectively.

## 3. Methods

This section presents the process to learn Comprehensive Global Embedding (CGE). We first describe the main framework of CGE. Next, the implementations of two modules including Non-parameterized Local Classifier (NLC) and Global Logits Revise (GLR) are introduced in detail. Finally, we briefly describe the loss functions of our method.

### *3.1. Overview*

The framework of Comprehensive Global Embedding (CGE) is shown in Figure 2. It mainly consists of five key components including Backbone, Pooling Operations, Classifier, Global Logits Revise (GLR) and Non-parameterized Local Classifier (NLC).

- Backbone. A pedestrian image passes through a CNN (*e.g.*, ResNet-50 [10]) to obtain feature maps, denoted as a tensor $F \in \mathbb{R}^{h \times w \times c}$.

- Pooling Operations. One of two candidate pooling strategies (Global Average Pooling (GAP) or Generalized-mean Pooling (GeM) [2]) is applied on $F$ to obtain a compact feature embedding $f_g \in \mathbb{R}^c$. AAP denotes adaptive average pooling which divides the feature maps into $n$ stripes and then conducts pooling on each stripe, resulting $n$ local features $\{f_i\}_{i=1}^n, f_i \in \mathbb{R}^c$. Note that when GeM is applied, the learned exponential parameter is used in AAP as well in the same way.

- Classifier. A fully-connected (FC) layer is used to transfer the feature embedding $f_g$ into the global logits $p \in \mathbb{R}^N$. Here $N$ is the number of training classes.

- Global Logits Revise. GLR module modifies the global logits, thereby changing the gradients, which enables samples with hard-to-learn local regions to have high weights while those with comprehensive global embeddings to have low weights.

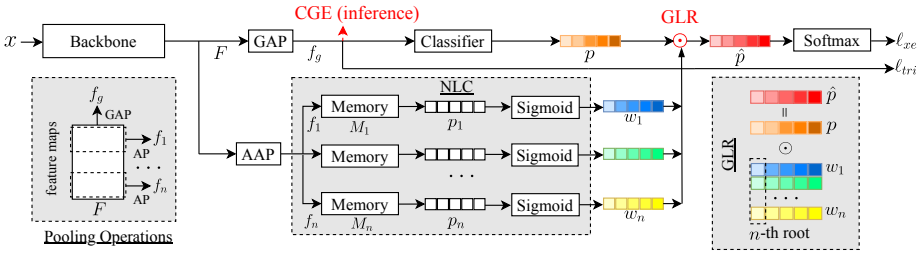



Figure 2: The framework of the proposed CGE. Given a person image $x$, it is fed into a backbone network, yielding the feature maps $F$. We obtain the global embedding $f_g$ via global average pooling (GAP) on $F$. 'AAP' means dividing $F$ into $n$ stripes and conducting average pooling on each stripe, resulting $n$ local features $\{f_i\}_{i=1}^n$. $f_g$ then goes through a classifier. The output is global logits $p$. $p$ is then revised by the GLR module. Another input of GLR module is $\{w_i\}_{i=1}^n$ which comes from the NLC module. Finally, after a Softmax function, we compute the cross entropy loss based on $\hat{p}$, as well as the triplet loss on $f_g$.

- Non-parameterized Local Classifier. NLC computes the discriminative abilities $\{w_i\}_{i=1}^n$ of $n$ local regions on $F$. The $\{w_i\}_{i=1}^n$ is then used in GLR module as weights to revise global logits $p$. The revised one is denoted as $\hat{p}$.

### *3.2. Non-parameterized Local Classifier*

The detailed structure of the proposed Non-parameterized Local Classifier(NLC) module is demonstrated in the middle grey box of Figure 2. Given the feature map $F$ computed by backbone network from an input pedestrian image, similar to some part-based methods, we split the initial $F$ into multiple horizontal stripes of size $h/n \times w \times c$. Here $n$ is the number of local regions. We apply pooling operation to each of them (e.g. GAP or GeM Pooling [2]), and obtain part-level features, denoted as $\{f_i\}_{i=1}^n, f_i \in \mathbb{R}^c$.

**Memory.** The memory $M$ is a feature bank [13] that stores the up-to-date features of all the classes in the dataset. Given a dataset including $N$ classes, we construct a memory $M$ which has $N$ slots. In particular, we design multiple part-level memories, denoted as $\{M_i\}_{i=1}^n, M_i \in \mathbb{R}^{c \times N}$. Each slot of $M_i$ stores the L2-normalized local feature of each class, whose dimension is $c$.

**Read Operation.** As the word 'Read' suggests, we fetch the local features of each class from the above-mentioned memory $M$, to compute the local score vectors. We name the memory $M$ as non-parameterized local classifiers. Specifically, given a local feature $f_i$ of sample $x$ with class $y$, we first compute the cosine similarities between it and features saved in the memory $M_i$. Then, we pass the similarity vector through a Sigmoid function to obtain the weight vector of $f_i$ which is normalized to $[0, 1]$, denoted as $w_i \in \mathbb{R}^N$. For $n$ regions, we obtain a set of weight vectors $\{w_i\}_{i=1}^n$. These weight vectors will be used in the Global Logits Revise (GLR) module.

**Write operation.** 'Write' here means to update the memory $M$. In the initialization, we initialize the values of all the features in the memories to zeros. For an input training sample $x$ with the class $y$, we obtain the L2-normalized local features $\{\|f_i\|_2\}_{i=1}^n, f_i \in \mathbb{R}^c$. During the back-propagation, we update the local features in the memory for the class $y$ through,

$$M_{i,y} \leftarrow \alpha * M_{i,y} + (1 - \alpha) * \|f_i\|_2 \tag{1}$$

where $M_{i,y}$ is the $i$-th local feature of class $y$ in the $y$-th slot of $M_i$. The hyper-parameter $\alpha \in [0, 1]$ controls the updating rate. $M_{i,y}$ is then L2-normalized via $M_{i,y} \leftarrow \|M_{i,y}\|_2$.

### *3.3. Global Logits Revise*

We aim to optimize the network to adaptively extract the comprehensive feature of a pedestrian. To this end, we apply the Global Logits Revise (GLR) module. It modifies the global logits by the weights computed from multiple local regions, aiming to make all of them discriminative. We first introduce the forward process of GLR and then its impact in a later paragraph.

**Forward.** As demonstrated in the right grey box in Figure 2, we modifies the global logits $p$ by $\{w_i\}_{i=1}^n$ and obtain $\hat{p}$ before passing $p$ to the loss function. Specifically, we merge $p$ with $n$ weight vectors $\{w_i\}_{i=1}^n$ of length $N$ via element-wise product. Each dimension of $w_i$ contains a scalar value between 0 and 1 via Sigmoid function. These vectors are obtained by the NLC, indicating the

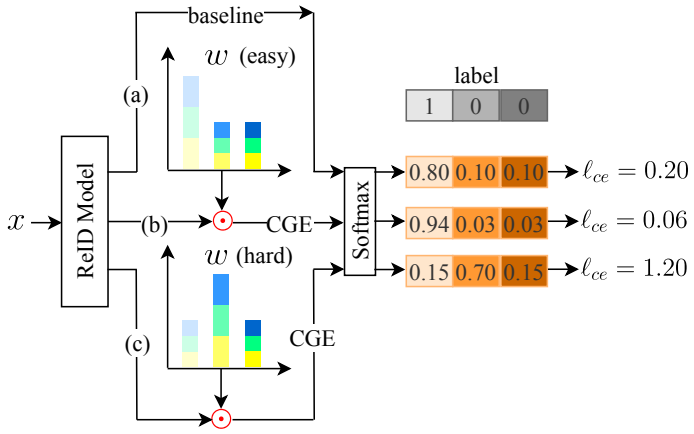



Figure 3: A toy example of the impact of CGE on the learning process. Three paths including (a), (b), (c) are shown in the figure. Path (a) denotes the baseline without any modification on the global logits. The final cross entropy loss is 0.20 in this toy example. Path (b) indicates the case where the input is an easy sample. 'Easy' here means the sample can be correctly classified by any of the single local feature. Thus the value of the true class dimension is enlarged and the cross entropy loss is reduced. Path (c) indicates the case where the input is a hard sample. Thus the value of the false class dimension is enlarged, leading to an increase in the cross entropy loss.

discriminative abilities over all classes of each region on feature map $F$. More specifically, to obtain the new global logits $\hat{p}$, we first compute the element-wise product $\odot$ between $n$ local weights and obtain $w$.

$$w = w_1 \odot w_2 \odot \cdots \odot w_n \tag{2}$$

Then we take the $n$-th root at each dimension of $w$ to prevent the value of $\hat{p}$ from being too small, which is not conducive to network optimization. Finally we merge the original global logits with this weight vector in the following function.

$$\hat{p} = p \odot w \tag{3}$$

**Impact of GLR.** This section discusses the loss values for samples of different hardness levels (easy/hard) and in different situations (baseline/CGE).

We use a toy example to explain the impact of CGE behind. As illustrated in Figure 3, follow the path (a), we derive that the cross entropy loss of baseline is 0.20, computed on the prediction and label. While following the path (b), we find the cross entropy loss much lower. The reason is that, for an easy sample, the local classifiers can identify the local region correctly, and obtain a high value on the dimension of the true class (the first dimension is higher in $w$). Correspondingly, $w$ increases the value of true class dimension of $p$, thereby reducing the loss. However, when following the path (c), the cross entropy loss is much higher compared with baseline (Path (a)). Since the input sample is a hard one, all local classifiers identify it into a wrong class. We can see that the second dimension of $w$ is the highest, which increases the value of wrong class dimension of $p$, thereby increasing the loss by a large margin.

To sum up, GLR module uses discriminative abilities of local regions on feature map to revise the global logits, thereby influencing the loss, which in turn increases the importance of correcting local regions that are mis-classified. GLR module thus affects the importance assignment of the backbone on samples during the training process.

### *3.4. Loss Functions*

We implement the CGE onto two baseline methods including BagTricks (BOT) [1] and AGW [2]. In these two versions, we adopt cross-entropy loss with label smoothing [33] and triplet loss with hard sample mining [15]. Note that Weighted Regularization Triplet (WRT) loss [34] is adopted in AGW [2]. Specifically, for each training sample $x$, our model computes the global logits (the revised one) of each label $k \in \{1, 2, \cdots, N\}$, denoted as $\hat{p} \in \mathbb{R}^N$. The identity label of $x$ is $y$, and the smoothed label is denoted as $\hat{y}$. The cross-entropy loss $\ell_{xe}(x)$ is computed as follows:

$$\ell_{xe}(x) = -\sum_{k=1}^{N} \hat{y_k} \log s(\hat{p_k}), \tag{4}$$

where $k$ denotes the $k$-th entry. $s(\cdot)$ denotes the Softmax function. $\hat{y}$ is the smoothed label, the $k$-th entry of $\hat{y}$ is computed as :

$$\hat{y_k} = (1-\epsilon)y_k + \frac{\epsilon}{N}, \tag{5}$$

where $\epsilon$ is a hyper-parameter and is often assigned as 0.1. $\ell_{xe}$ is obtained by summing the losses of all training samples.

Secondly, we introduce the triplet loss with hard sample mining for the global feature $f_g$. We compute the triplet loss as follows:

$$\ell_{tri} = \frac{1}{PK}\sum_{i=1}^{PK}[m + D(f_{g,i}^a, f_{g,i}^+) - D(f_{g,i}^a, f_{g,i}^-)]_+, \tag{6}$$

where $[]_+ = max(\cdot, 0)$ and $m$ is the margin between the positive and negative pairs. Specifically, we form batches by randomly sampling $P$ labels, and then randomly sampling $K$ images of each label, thus resulting in a batch of $P * K$ images. For each sample in the batch, we select the hardest positive and the hardest negative sample within the batch to form the triplets for computing the loss. The representations of the $i$-th triplet is denoted as $(f_{g,i}^a, f_{g,i}^+, f_{g,i}^-)$. $D$ is the Euclidean distance and is used to compute the distance among feature representations. In our experiments, hyper-parameters are set as follows. $m$ is set to 0. We randomly sample $P=16$ labels, and $K=4$ images for each selected label to form a mini-batch. Note that since we use the $PK$ sampling, when we conduct the write operation on the memories (mentioned in Section 3.2) in the NLC module, we first compute the average feature of $K$ instances, and then use the average one to update the memory.

Thirdly, we introduce the Weighted Regularization Triplet (WRT) loss. We compute the WRT loss as follows:

$$\ell_{wrt} = \frac{1}{PK}\sum_{i=1}^{PK}[\log(1+\exp(\lambda^+ D(f_{g,i}^a, f_{g,i}^+) - \lambda^- D(f_{g,i}^a, f_{g,i}^-)))] \tag{7}$$

$$\lambda^+ = \frac{\exp(D(f_{g,i}^a, f_{g,i}^+)}{\sum_{f_g^+ \in \mathcal{P}} \exp(f_g^+)}, \lambda^- = \frac{\exp(D(f_{g,i}^a, f_{g,i}^-)}{\sum_{f_g^- \in \mathcal{N}} \exp(f_g^-)} \tag{8}$$

where $\mathcal{P}$ is the positive set of anchor image and $\mathcal{N}$ is the negative set.

Finally, we train the CGE end-to-end with the following objective function in the BagTricks (BOT) [1] version:

$$\ell = \lambda_1 \ell_{xe} + \lambda_2 \ell_{tri} \tag{9}$$

The objective function in the AGW [2] version is as follows:

$$\ell = \lambda_1 \ell_{xe} + \lambda_2 \ell_{wrt} \tag{10}$$

where $\lambda_1, \lambda 2$ are two hyper-parameters to assign different weights to the cross entropy loss $\ell_{xe}$ and $\ell_{tri}/\ell_{wrt}$.

## 4. Experiments

We run numerous experiments on three Re-ID benchmarks in order to validate the effectiveness of the proposed CGE. In this section, we first introduce the benchmarks and the implementation details. Then we conduct ablation studies to analyze the behavior of CGE as well as evaluating the influence of all the hyper-parameters. After that, we show some visualization results to obtain qualitative analysis results. Finally, we compare our proposed CGE with other state-of-the-arts methods.

### *4.1. Datasets and Settings*

#### *4.1.1. Datasets*

We evaluate our approach on three widely used Re-ID datasets, *i.e.*, Market-1501 [35] CUHK03 [36] and MSMT17 [37].

- **Market-1501** [35] is collected from six different camera views. It has $32,668$ bounding boxes of $1,501$ identities obtained using a Deformable Part Model (DPM) person detector. Following the standard splits, we use $12,936$ images of 751 identities for training and $19,281$ images of 750 identities in the gallery for testing.

- **MSMT17** [37] consists of $4,101$ pedestrians with $126,441$ images captured by 15 cameras. The training set consists of $1,041$ people, containing

32,621 images, and the test set consists of 3,060 people, containing 93,820 images.

- **CUHK03** [36] is composed of 1,467 pedestrians with 14,097 images captured by two of ten cameras. The training set consists of 767 people, containing 7,365 images, and the test set consists of 700 people, containing 1,400 query images and 5,332 gallery images. CUHK03 provides manually-labeled and DPM-detected bounding boxes for all images. We denote the former as 'CUHK03-L' and the latter as 'CUHK03-D'.

#### *4.1.2. Evaluation Metrics*

Following related works [38], we use the Cumulative Matching Characteristic (CMC) curve and the Mean Average Precision (mAP) for performance evaluation. For each query, its average precision (AP) is computed from its precision-recall curve. The mAP is calculated as the mean value of average precision across all queries. We report the rank1 score to represent the CMC curve. Rank1 score reflects the retrieval precision, while the mAP reflects the recall.

#### *4.1.3. Implementation Details*

We will present the implementation details including the backbone model, pre-processing, training settings and hyper-parameters in this section.

- **Model.** Fu *et al.* proved the effectiveness of unsupervised pre-training in Re-ID [39]. Since we want CGE to work on high-performing baselines, we use a ResNet-50 [10] pre-trained on LUPerson (a large scale Re-ID dataset presented in [39]) with the modified version of MoCo-v2 proposed in [39] as our backbone network. To make a fair comparison with existing methods, we also report the results of adopting the ResNet-50 pre-trained on ImageNet[25] as our backbone. We set the stride of last residual module convolution operation to 1 to obtain higher-resolution feature maps [6].

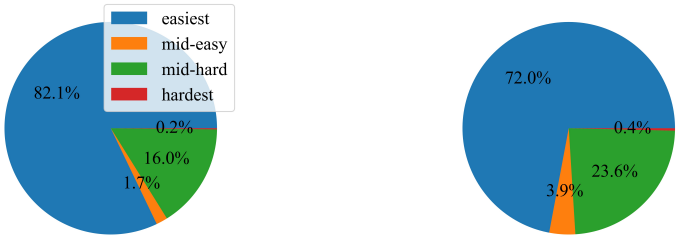
- **Pre-processing.** All input images to the backbone are resized as $384 \times 128$. Random flipping, random cropping and random erasing [11] are employed as the data augmentation for training stage.

- **Training settings.** We set the mini-batch size to 64. Each identity contains 4 instance images, so there are 16 identities per batch. All experiments use the Adam optimizer with the base learning rate initialized to $3.5 \times 10^{-5}$. With a linear warm-up strategy in first 10 epochs, the learning rate increases to $3.5 \times 10^{-4}$. Then, the learning rate is decayed to $3.5 \times 10^{-5}$ after 40 epochs, and further decayed to $3.5 \times 10^{-6}$ after 90 epochs. The whole training procedure has 120 epochs. Our network is trained using one Tesla V100 GPU.

- **Hyper-parameters.** There are multiple hyper-parameters in the proposed method, *i.e.,* $n$ local regions, $\alpha \in [0, 1]$ in Equation (1) controlling the updating rate, $s_h$ indicating the strength of REA, $\epsilon$ denoting the strength of label smoothing and $\lambda_1, \lambda_2$ controlling the weights between cross entropy loss and triplet loss. According to our ablation study, setting $n = 3$, $s_h = 0.4$, $\epsilon = 0.1$, $\lambda_1 = 1$, $\lambda_2 = 1$ achieves the best performance, which will be described in detail in the next sub-section. The important hyper-parameter $\alpha$ in Equation 1 is determined according to an linear function $\alpha = \frac{1}{E} * t$, where $t$ denotes the current epoch and $E$ denotes the entire training epochs. The ablation study on $\alpha$ will be described in detail in the next sub-section as well.

### *4.2. Ablation Study*

In this section, we first analyze the proposed CGE via judging the complexity of the benchmarks as well as comparing with the well-known focal loss [12]. Then we conduct numerous experiments on the hyper-parameters, enabling CGE to play its best role. Finally, we conduct ablation study on multiple top-performing Re-ID loss functions as well we whether to revise the feature or logits in the GLR module.

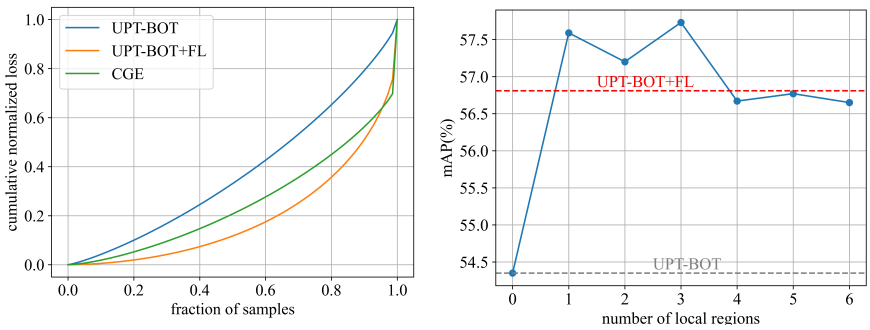



(a) Market-1501 (b) MSMT17

Figure 4: Pie charts of four-level samples on the train sets of two benchmarks. We set $n = 3$.

*4.2.1. Analysis of the CGE*

**CGE improves more on complex datasets.** How to define whether a dataset is simple or complex? To answer this question, we divide the training samples of two benchmarks into four levels, *i.e.*, the easiest, mid-easy, mid-hard and hardest. Specifically, we use the maintained local memory banks $\{M_i\}_{i=1}^{3}$ to classify the samples based on the corresponding local region. If the sample can be correctly classified according to all single local regions, we will regard it as an easiest one. Apparently, if the sample cannot be correctly classified by any local region, we regard it as a hardest one. The mid-easy ones are defined as samples can be correctly classified by any two local regions and the mid-hard ones are those can be correctly classified by one local region. We compute the proportion of these four-level samples on Market-1501 and MSMT17. The pie charts are shown in Figure 4. According to Figure 4, the proportions of the easiest samples are 82.1% and 72.0% respectively on two benchmarks. The corresponding improvements on mAP%/rank-1% of CGE compared with baseline UPT-BOT are 2.4%/0.6% and 5.3%/3.8% (shown in Table 9). The experimental results are reasonable. Since when easy samples play a leading role, the image can be correctly matched based on any single local feature. It is impossible to highlight the important impact of the comprehensiveness of the global feature on Re-ID




(a) (b)

Figure 5: Left: Cumulative distribution functions of the normalized loss for training samples on MSMT17. 'UPT-BOT' denotes the BOT [1] model pre-trained by UPT [39]. 'FL' means focal loss [12] and 'CGE' is the proposed method. Right: Ablation study on the number of local regions $n$ on MSMT17. The red dash line denotes the mAP(%) of UPT-BOT+FL and the grey dash line denotes the mAP(%) of baseline.

performance.

**Comparison with focal loss.** To understand the proposed CGE better, we analyze the empirical distribution of the loss as well as the prediction results of three converged models: the baseline model UPT-BOT [1], UPT-BOT+FL (FL means focal loss [12]) and the proposed CGE.

For the train set of MSMT17, we compute the cross entropy loss of UPT-BOT, the focal loss of UPT-BOT+FL and the revised cross entropy loss (when $n = 3$) of CGE. We normalize each loss such that it sums to one. Given the normalized loss, we sort the loss from the lowest to the highest and plot the cumulative distribution function (CDF) respectively, according to [12]. Cumulative distribution functions are shown in Figure 5a. Firstly, we compare the blue and orange line in Figure 5a. As stated in [12], focal loss (FL) makes the model pay more attention to hard samples, since the vast majority of the loss comes from a small fraction of samples. The same phenomenon can be found by comparing the blue and green line, which validates the effectiveness of the proposed CGE. When comparing the green and orange lines, we can draw some

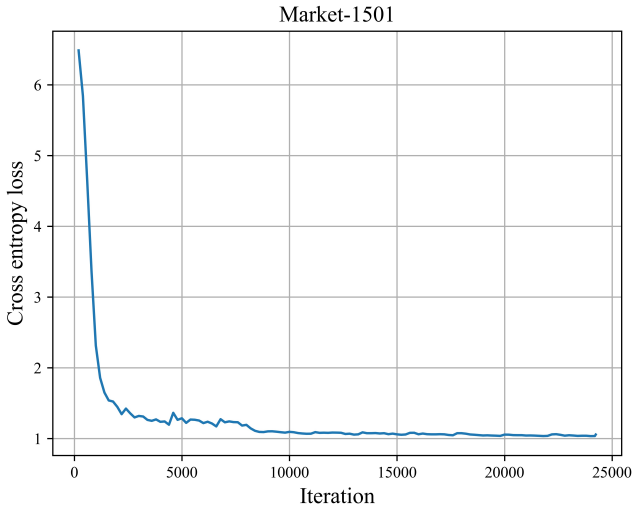



Figure 6: The training loss of CGE when $n = 12$ on Market-1501.

interesting conclusions. These two lines have a staggered point. This means that the so-called hard samples in focal loss contain both the mid-hard and hardest samples (illustrated by Figure 4), indicating that FL may not be able to distinguish between these two types of samples. However, CGE assigns more weight to the hardest samples than the mid-hard ones. The phenomenon that the vast majority of the loss comes from a small fraction of samples is more obvious in CGE than in focal loss. In other words, CGE makes a more fine-grained division of the importance of samples than focal loss. Note that though with larger $\gamma$ in focal loss, the green line will be more towards the lower right corner of the coordinate system, better Re-ID performance is not guaranteed. According to our numerous experiments, when $\alpha = 0.5, \gamma = 1.0$, focal loss performs the best. The mAP and rank-1 are 56.81% and 79.08%, the improvement over baseline is 2.46% and 1.39%; thus we draw the green curve with this set of hyper-parameters.

| | | |
|---|---|---|
| n=12 | mAP | 88.5% |
| | rank-1 | 95.3% |
| n=6 | mAP | 89.6% |
| | rank-1 | 95.8% |
| n=3 | mAP | 90.8% |
| | rank-1 | 95.9% |

(a) varying $n$ for CGE

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| n=12 | $M_1$ | 22.2% | $M_4$ | 1.6% | $M_7$ | 0.1% | $M_{10}$ | 0.1% |
| | $M_2$ | 55.0% | $M_5$ | 10.4% | $M_8$ | 0.0% | $M_{11}$ | 0.0% |
| | $M_3$ | 86.3% | $M_6$ | 27.4% | $M_9$ | 0.1% | $M_{12}$ | 0.0% |
| n=6 | $M_1$ | 50.2% | $M_4$ | 1.9% | | $M_1$ | 96.6% | |
| | $M_2$ | 93.5% | $M_5$ | 3.6% | n=3 | $M_2$ | 97.1% | |
| | $M_3$ | 87.6% | $M_6$ | 0.9% | | $M_3$ | 91.9% | |

(b) precision of $\{M_i\}_{i=1}^{n}$ on the train set

Table 1: **Ablation experiments for CGE on Market-1501.** If not specified, default backbone is ResNet-50 pretrained by [39].

*4.2.2. Impact of the hyper-parameters*

**Varying the number of local regions.** The proposed CGE introduces one new hyper-parameter *i.e.*, the number of local regions $n$ on feature maps. $n$ depicts the degree of sophistication in the importance assignment process. The ablative results are shown in Figure 5b, Table 1 and Table 2. According to Figure 5b, when $n = 1, 2, 3$, the performances of CGE are all superior to focal loss. CGE performs best when $n = 3$. However, when $n = 4, 5, 6$, the accuracy drops. We suppose the reason is that there are no discriminative clues within a very small stripe of some samples, misleading the network to assign them with high weights. To validate the supposition, we compute the precision of each non-parameterized local classifier, *i.e.*, $\{M_i\}_{i=1}^{n}$. The results are shown in Table 1b. When $n = 12$, the precision of some local classifiers is inaccurate, *e.g.*, $M_{10}, M_{11}, M_{12}$, which is close to 0. These local logits are not helpful to revise the global logits. Instead, they mislead the backbone to assign wrong weights to samples. The ablative results of $n$ on Market-1501 are shown in Table 1a. We can see that when $n$ is too large, *e.g.*, $n = 12$, the Re-ID performance is inferior, though the model converges even when $n = 12$ (shown in Figure 6). On Market-1501, we achieve the best performance when $n = 3$ as well.

To validate that the local regions on global feature maps of CGE are more discriminative than those of the baseline, we report the comparison results of precision on the most complex benchmark, *i.e.*, MSMT17. The results are shown

| Local classifier | CGE | CGE wo. GLR |
|---|---|---|
| $M_1$ | 97.8% | 97.2% |
| $M_2$ | 92.3% | 91.6% |
| $M_3$ | 82.0% | 79.5% |

Table 2: Precision of $\{M_i\}_{i=1}^3$ on the train set of MSMT17.

| $s_h$ | mAP% | Rank-1% |
|---|---|---|
| 0.1 | 53.59 | 79.11 |
| 0.2 | 56.41 | 80.13 |
| 0.3 | 57.34 | **80.34** |
| 0.4 | **57.73** | 80.08 |
| 0.5 | 56.05 | 78.74 |
| 0.6 | 55.58 | 77.43 |

(a) varying $s_h$ for REA [11]

| $\epsilon$ | mAP% | Rank-1% |
|---|---|---|
| 0.00 | 56.73 | 78.97 |
| 0.05 | 57.66 | 79.52 |
| 0.10 | **57.73** | **80.08** |
| 0.15 | 56.01 | 79.08 |
| 0.20 | 54.87 | 78.39 |
| 0.25 | 54.65 | 78.51 |

(b) varying $\epsilon$ for label smoothing [33]

| $\lambda 1$ | $\lambda 2$ | mAP% | Rank-1% |
|---|---|---|---|
| 1.0 | 1.0 | **57.73** | **80.08** |
| 0.9 | 0.1 | 56.13 | 79.24 |
| 0.8 | 0.2 | 57.10 | 79.79 |
| 0.7 | 0.3 | 57.29 | 79.80 |
| 0.6 | 0.4 | 56.95 | 79.74 |
| 0.4 | 0.6 | 56.55 | 79.20 |

(c) varying $\lambda_1, \lambda_2$ for $\ell_{xe}$, $\ell_{tri}$

Table 3: **Ablation experiments for CGE on MSMT17.** If not specified, default backbone is ResNet-50 pretrained by [39]. (a) The strength of REA (via adjusting the value of $s_h$) has an impact on the performance of CGE. Other hyper-parameters in REA are: $p = 0.5, s_l = 0.02, r_e = 0.3$. (b) The strength of label smoothing also influences CGE. $\epsilon$ denotes the tiny weights assigned to false classes. (c) Different weights of the cross entropy loss and triplet loss.

in Table 2. We can see that the discriminative ability of the third local region is improved in CGE.

**Varying the strength of random erasing.** The performance of CGE is influenced by the strength of random erasing. If we replace too many regions on images with random values, the local logits will mislead the backbone as well. However, REA is effective in Re-ID since it eliminates the potential occlusion problem during inference. The trade-off between its advantage and disadvantage should be well balanced via tuning the hyper-parameter $s_h$ in REA. The ablative results are reported in Table 3a. We can see that when $s_h$ is set to 0.4, the model achieves the best mAP.

**Varying the strength of label smoothing.** The performance of CGE is

| $\alpha$ | Market-1501 | | MSMT17 | |
|---|---|---|---|---|
| | mAP% | Rank-1% | mAP% | Rank-1% |
| $\frac{1}{E} * t$ | **90.8** | **95.9** | **59.7** | **81.5** |
| $e^{-5*(\frac{1-t}{E})^2}$ | 89.8 | 95.8 | 57.7 | 80.1 |
| 0.01 | 89.7 | 95.4 | 57.1 | 79.6 |

Table 4: The ablative results of $\alpha$ in Equation 1 on Market-1501 and MSMT17.

influenced by the strength of label smoothing. Label smoothing is designed to prevent the network from making over-confident predictions on the true class. It assigns small values which is denoted as $\epsilon$ to the dimensions of the false classes. The ablative results of varying $\epsilon$ for label smoothing is shown in Table 3b. We can see that $\epsilon = 0.10$ shows both the best mAP and rank1 accuracy.

**The effect of $\lambda_1, \lambda_2$.** The balance coefficients $\lambda_1, \lambda_2$ in Equation 9 and Equation 10 control the importance of the cross entropy loss (with revised global logits in the proposed GLR module) and the triplet loss. The ablative results on $\lambda_1, \lambda_2$ are shown in Table 3c. We set different values for $\lambda_1, \lambda_2$, to evaluate the performance of the proposed CGE. We can see that when $\lambda_1 = 1, \lambda_2 = 1$, the model achieves the best performance.

**The effect of $\alpha$.** The $\alpha$ in Equation 1 controls the updating rate, which is an important hyper-parameter. We report the results of three different strategies. The first strategy is to linearly increase the $\alpha$ as follows:

$$\alpha = \frac{1}{E} * t \tag{11}$$

where $t$ denotes the current epoch, and $E$ denotes the number of total training epochs. The second strategy is to exponentially decay the $\alpha$ as follows:

$$\alpha = e^{-5*(\frac{1-t}{E})^2} \tag{12}$$

The third strategy is to fix the $\alpha$ as 0.01. The ablative results are shown in Table 4. We can see from Table 4 that the first strategy achieves the best performance. Thus the we adopt it in all our following experiments.

| Loss | BOT [1] | | CGE-BOT (proposed) | |
|---|---|---|---|---|
| | mAP% | rank1% | mAP% | rank1% |
| CE | 85.2 | 94.4 | 87.2 (↑2.0) | 95.0 (↑0.6) |
| CE+Sphere [45] | 85.0 | 94.2 | 86.0 (↑1.0) | 94.5 (↑0.3) |
| CE+Triplet [15] | 86.3 | 94.4 | 88.0 (↑1.7) | 95.0 (↑0.6) |
| CE+Lifted [41] | 87.0 | 94.2 | 88.0 (↑1.0) | 94.6 (↑0.4) |
| CE+Instance [42] | 85.2 | 94.0 | 87.6 (↑1.4) | 95.0 (↑1.0) |
| CE+Contrast [43] | 85.9 | 94.0 | 87.3 (↑1.4) | 94.5 (↑0.5) |
| CE+Circle [44] | 87.1 | 94.1 | 88.4 (↑1.3) | 94.9 (↑0.8) |

Table 5: The ablative results of loss functions on Market-1501. 'CE' here denotes the cross entropy loss with label smoothing.

#### *4.2.3. Impact of the loss functions*

To validate whether the proposed method is compatible with some top-performing loss functions in Re-ID. We conduct ablation study on loss functions including Sphere [40], Lifted [41], Instance [42], Contrast [43] and Circle [44] loss, on the BOT baseline and the proposed CGE baseline. Note that the hyper-parameter in Sphere loss is: $m = 4$, and those in Circle loss are: $\gamma = 2, m = 0.25$. The results are reported in Table 5.

We can see that the proposed CGE improves the mAP of all the models by more than 1%. These results validate that CGE is well compatible with existing top-performing loss functions in Re-ID.

#### *4.2.4. Feature-level or logits-level*

Since we revise the global logits instead of the feature, we suspect whether feature-level revise can also take effect. To conduct feature-level revise, we add a convolutional layer to convert the channel of the output feature from ResNet-50 to the number of classes (e.g. 751 on Market-1501). Then we use the weights from the NLC module to revise each feature map, followed by global average pooling on each feature map. The comparison between feature-level and logits-level revise is shown in Table 6. We can see the improvement brought by revising the feature is inferior to that brought by refining the global logits.

| Methods | BOT [1] | | CGE-BOT (proposed) | |
|---|---|---|---|---|
| | mAP% | rank1% | mAP% | rank1% |
| Feature | 83.0 | 92.6 | 83.8 (↑0.8) | 92.9 (↑0.3) |
| Logits | 85.9 | 94.5 | 88.0 (↑2.1) | 95.0 (↑0.5) |

Table 6: The comparison between revising the feature and global logits on Market-1501.

### *4.3. Qualitative analysis*

We compare the Saliency maps of the global image representation of UPT-BOT [1] and CGE on three benchmarks in Figure 7. Comparing the second and the third column in Figure 7, we can see that naive global features are trapped in the most discriminative local region, reflected in that only a single location has large values. Note that the Saliency maps are drawn via taking the maximum of feature maps along the channel dimension [9]. Larger activation values mean more contribution to the loss. However, the CGE is better, manifested in that is has multiple hot areas which are highly activated.

### *4.4. Scalability*

We suspect the proposed CGE can be extended to other retrieval tasks such as vehicle re-identification which also focuses on shape; thus we conduct experiments on the vehicle re-identification benchmark: **VeRi**.

**VeRi** contains over $50,000$ images of 776 vehicles captured by 20 cameras. The training set consists of 576 vehicles, containing $37,778$ images, and the test set consists of 200 vehicles, containing $1,678$ query images and $11,579$ gallery images.

In terms of the baseline, we adopt the BagTricks (BOT) [1]. The pre-processing, training settings and hyper-parameters are the same with those adopted in Re-ID task. The results are shown in Table 7. We can see that the proposed CGE is also effective in the vehicle re-identification task. We may infer that the proposed CGE will be effective in some other shape-related retrieval tasks.

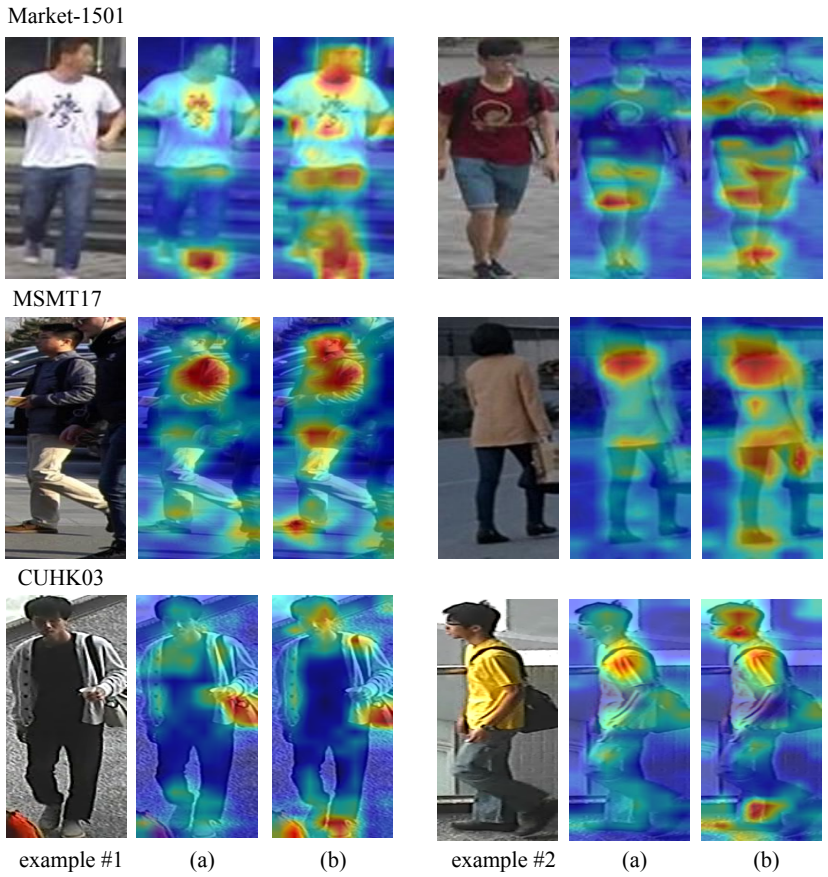



Figure 7: The saliency maps of global image representation $F$. The backbone is trained by (a) UPT-BOT [1]; (b) CGE (ours)

### *4.5. Comparison with State-of-the-Arts Methods*

Table 9 compares our CGE method to the state-of-the-arts (SOTA) respectively on three benchmarks including Market-1501, MSMT17 and CUHK03. The table is divided into two blocks *i.e.*, Local and Global. We mainly compare the proposed CGE with global representation learning methods. Our CGE is a general training approach, and we plug it in some of the SOTA baselines including BagTricks (BOT) [1] and AGW [2] to validate its efficiency and generality. Note that we enhance BagTricks (BOT) and AGW with unsupervised pre-training [39] and term the model as UPT-BOT and UPT-AGW (Table 9 in

| Methods | mAP% | rank1% |
|---|---|---|
| BOT | 73.9 | 93.6 |
| BOT+CGE (proposed) | 76.2 (↑2.3) | 94.9 (↑1.3) |

Table 7: The scalability of the proposed CGE on vehicle re-identification benchmark: VeRi [46].

| Training details | Ye *et al.* [2] | Fu *et al.* [39] |
|---|---|---|
| image size | $256 \times 128$ | $384 \times 128$ |
| lr decay steps | [40,70] | [40,90] |
| BNNeck [1] feature | after | before |

Table 8: The difference of AGW implemented by Ye *et al.* [2] and Fu *et al.* [39] on MSMT17.

bold). We also plug CGE in the ImageNet pre-trained BagTricks (BOT) and AGW to make fair comparisons.

**Market-1501** As shown in Table 9, on Market-1501, when adopting the ImageNet pre-trained BOT and AGW, the CGE versions achieve 2.1% and 2.3% mAP improvements respectively. On the high-performing baseline, *i.e.*, UPT-BOT and UPT-AGW, the CGE versions achieve 2.4% and 0.4% mAP improvements respectively.

**MSMT17** As for MSMT17 which is the largest benchmark, the improvements are larger. CGE gets 4.1% better than BOT and 6.7% better than AGW on mAP. As for UPT-BOT and UPT-AGW, the improvements on mAP are 5.3% and 4.0% respectively. Note that the performance of AGW re-implemented by us on MSMT17 is superior to the that of the original version. Our experiments are based on the code released by Fu *et al.* [39]. We show the differences between this version and the original one in Table 8.

**CUHK03** As shown in Table 9, on both the CUHK03 labeled and detected datasets, the proposed CGE improves the baselines by a large margin. CGE achieves state-of-the-arts performance on CUHK03 via only adopting the global feature.

Table 9: Comparison with state-of-the-arts methods on Market-1501, MSMT17 and CUHK03. The results are shown in mAP%/rank1%. The red font represents the best performance in the corresponding block while the blue font denotes the second best. The Bold indicates the baseline models and the Italics are our re-implemented results. '-BNNeck' denotes the novel structure proposed by Luo *et al.* [1].

| | Methods | Backbone | Market-1501 | MSMT17 | CUHK03-L | CUHK03-D |
|---|---|---|---|---|---|---|
| Local | PCB [6] (2018) | ResNet-50 | 81.6/93.8 | - | - | 57.5/63.7 |
| | Auto [47] (2019) | Searched | 85.1/94.5 | 52.5/78.2 | 73.0/77.9 | 69.3/73.3 |
| | MGN [8] (2018) | ResNet-50 | 86.9/95.7 | - | 67.4/68.0 | 66.0/66.8 |
| | st-ReID [21] (2019) | ResNet-50 | 87.6/98.1 | - | - | - |
| | DSA [23] (2019) | ResNet-50 | 87.6/95.7 | - | | |
| | SAN [24] (2020) | ResNet-50 | 88.0/96.1 | 55.7/79.2 | 76.4/80.1 | 74.6/78.4 |
| | ISP [7] (2020) | HRNet [48] | 88.6/95.3 | - | 74.1/76.5 | 71.4/75.2 |
| | LFS-ReID [49] (2022) | ResNet-50 | 89.4/95.8 | 62.3/82.6 | - | 73.6/76.9 |
| | UPT-MGN [39] (2021) | ResNet-50 | 91.0/96.4 | 65.7/85.5 | 74.7/75.4 | - |
| | VA-reID [20] (2020) | SeResNext | 91.7/96.2 | - | - | - |
| | HLGAT [18] (2021) | ResNet-50 | 93.4/97.5 | 73.2/87.2 | - | 80.6/83.5 |
| | PGCN [19] (2021) | ResNet-50 | 94.8/98.0 | - | 83.6/86.7 | - |
| Global | OSNet [50] (2019) | OSNet [50] | 84.9/94.8 | 52.9/78.7 | - | 67.8/72.3 |
| | **BOT** [1] (2019) | ResNet-50-BNNeck | 85.9/94.5 | *53.3/77.0* | *65.0/66.5* | *62.7/65.6* |
| | BDB [26] (2019) | ResNet-50 | 86.7/95.3 | - | 76.7/79.4 | 73.5/76.4 |
| | **AGW** [2] (2021) | ResNet-50-BNNeck | 87.8/95.1 | *57.8/81.1* | *73.0/74.6* | *69.7/71.1* |
| | DAAF-BoT [32] (2022) | ResNet-50 | 87.9/95.1 | - | 67.6/69.0 | 63.1/64.9 |
| | UPT-BDB [39] (2021) | ResNet-50 | 88.1/95.3 | 52.5/79.1 | 79.6/81.9 | - |
| | BOT [1] (2019) | IBN-Net50-a | 88.2/95.0 | - | - | - |
| | **UPT-BOT** | ResNet-50-BNNeck | *88.4/95.3* | *54.4/77.7* | *76.3/77.5* | *73.6/74.2* |
| | SCSN [28] (2020) | ResNet-50 | 88.5/95.7 | 58.5/83.8 | 84.0/86.8 | 81.0/84.7 |
| | SONA [51] (2019) | ResNet-50 | 88.8/95.6 | - | 79.2/81.4 | 77.3/79.9 |
| | SCAL [52] (2019) | ResNet-50 | 89.3/95.8 | - | 72.3/74.8 | 68.6/71.1 |
| | MEMF [30] (2021) | ResNet-50 | 89.5/96.1 | 59.8/82.9 | 73.6/76.7 | 70.9/74.1 |
| | SDN [31] (2022) | ResNet-50 | 89.6/95.8 | - | 77.2/80.1 | 74.5/76.9 |
| | **UPT-AGW** | ResNet-50-BNNeck | *91.8/96.2* | *61.9/82.2* | *82.3/83.2* | *79.0/79.8* |
| | CGE(ours)+ BOT | ResNet-50-BNNeck | 88.0/95.0 (↑2.1/0.5) | 57.4/80.5 (↑4.1/3.5) | 74.1/76.7 (↑9.1/10.2) | 70.6/72.8(↑7.9/7.2) |
| | CGE(ours)+UPT-BOT | ResNet-50-BNNeck | 90.8/95.9 (↑2.4/0.6) | 59.7/81.5 (↑5.3/3.8) | 82.7/84.0 (↑6.4/6.5) | 79.9/81.9 (↑6.3/7.7) |
| | CGE(ours)+ AGW | ResNet-50-BNNeck | 90.1/95.6 (↑2.3/0.5) | 64.5/83.9 (↑6.7/2.8) | 78.1/79.8 (↑5.1/5.2) | 75.0/77.4 (↑5.3/6.3) |
| | CGE(ours)+UPT-AGW | ResNet-50-BNNeck | 92.2/96.3 (↑0.4/0.1) | 65.9/85.1 (↑4.0/2.9) | 86.4/87.4 (↑4.1/4.2) | 84.2/85.9 (↑5.2/7.6) |

## 5. Conclusions

Naive global features tend to being trapped in the most discriminative local region. To tackle this problem, we design a novel network to ensure more local regions on the global feature maps of each sample to be discriminative. To realize this, two novel modules are proposed to up-weight the samples within hard-to-learn local regions. We found that the proposed method is especially effective on complex datasets such as MSMT17 and CUHK03. We improve the baseline mAP on MSMT17 and CUHK03 labeled by 4.0% and 4.1% respectively. The mechanism behind is that we revise the global logits according to the local discriminativeness, which increases the value of true class dimension for an easy sample thus decreasing the loss and increases the value of false class dimension for a hard sample thus increasing its importance. The converge of the model indicates more local regions are highly activated.

The strength of this method is: it imposes the constraints on the comprehensiveness of global feature into the training process. During inference, the backbone as well as the pedestrian embedding is succinct.

The weakness of this method is: the optimal number of local regions is three according to the experimental results, since we cannot guarantee the precision of non-parameterized local classifiers when the number of local regions becomes larger. By the way, we validate the comprehensiveness of the global feature via qualitative analysis; quantitative improvement need to be confirmed.

In the future, we plan to explore effective strategies to measure the importance of more fine-grained local regions (*i.e.*, pixel-wise, group of pixels), aiming to further improve the comprehensiveness of the global feature. We speculate that embedding some beneficial properties into the training process of pedestrian representation is an encouraging future direction, which will lead to succinct reference process.

To sum up, our work in this paper emphasizes on increasing the comprehensiveness of global features via enlarging the weights of samples within hard-to-learn local regions. The proposed method achieves state-of-the-arts performance

on multiple Re-ID benchmarks with only global features during inference. We hope that more works will be conducted based on our CGE to further improve the performance of retrieval tasks, since the global features learnt by CGE is more comprehensive.

**Acknowledgment**

This research is supported by A*STAR under its AME YIRG Grant (Project No. A20E6c0101), the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant, NSFC Projects (62071292, U21B2013).

**References**

[1] H. Luo, W. Jiang, Y. Gu, F. Liu, X. Liao, S. Lai, J. Gu, A strong baseline and batch normalization neck for deep person re-identification, IEEE Transactions on Multimedia 22 (10) (2020) 2597–2609. doi:10.1109/TMM.2019.2958756.

[2] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, S. C. Hoi, Deep learning for person re-identification: A survey and outlook, IEEE Transactions on Pattern Analysis and Machine Intelligencedoi:10.1109/tpami.2021.3054775.

[3] S. Liao, Y. Hu, X. Zhu, S. Z. Li, Person re-identification by local maximal occurrence representation and metric learning, in: CVPR, 2015, pp. 2197–2206. doi:10.1109/CVPR.2015.7298832.

[4] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: CVPR, 2012, pp. 2288–2295. doi:10.1109/CVPR.2012.6247939.

[5] X. Bai, M. Yang, T. Huang, Z. Dou, R. Yu, Y. Xu, Deep-person: Learning discriminative deep features for person re-identification, Pattern Recognition 98 (2020) 107036. doi:10.1016/j.patcog.2019.107036.

[6] Y. Sun, L. Zheng, Y. Yang, Q. Tian, S. Wang, Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline), in: ECCV, 2018, pp. 480–496. doi:10.1109/CVPR.2019.00048.

[7] K. Zhu, H. Guo, Z. Liu, M. Tang, J. Wang, Identity-guided human semantic parsing for person re-identification, in: ECCV, 2020, pp. 346–363. doi:10.1007/978-3-030-58580-8_21.

[8] G. Wang, Y. Yuan, X. Chen, J. Li, X. Zhou, Learning discriminative features with multiple granularities for person re-identification, in: ACMMM, 2018, pp. 274–282. doi:10.1145/3240508.3240552.

[9] H. Yao, S. Zhang, R. Hong, Y. Zhang, C. Xu, Q. Tian, Deep representation learning with part loss for person re-identification, IEEE Transactions on Image Processing 28 (6) (2019) 2860–2871. doi:10.1109/TIP.2019.2891888.

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[11] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: AAAI, 2020, pp. 13001–13008. doi:10.1609/aaai.v34i07.7000.

[12] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: ICCV, 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.324.

[13] Z. Zhong, L. Zheng, Z. Luo, S. Li, Y. Yang, Invariance matters: Exemplar memory for domain adaptive person re-identification, in: CVPR, 2019, pp. 598–607. doi:10.1109/CVPR.2019.00069.

[14] L. Zheng, Y. Yang, A. G. Hauptmann, Person re-identification: Past, present and future, arXiv:1610.02984.

[15] A. Hermans, L. Beyer, B. Leibe, In defense of the triplet loss for person re-identification, arXiv:1703.07737.

[16] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: CVPR, 2018, pp. 7794–7803. doi:10.1109/CVPR.2018.00813.

[17] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, Q. Tian, Pose-driven deep convolutional model for person re-identification, in: ICCV, 2017, pp. 3960–3969. doi:10.1109/ICCV.2017.427.

[18] Z. Zhang, H. Zhang, S. Liu, Person re-identification using heterogeneous local graph attention networks, in: CVPR, 2021, pp. 12136–12145. doi:10.1109/cvpr46437.2021.01196.

[19] Z. Zhang, H. Zhang, S. Liu, Y. Xie, T. S. Durrani, Part-guided graph convolution networks for person re-identification, Pattern Recognition 120 (2021) 108155. doi:10.1016/j.patcog.2021.108155.

[20] Z. Zhu, X. Jiang, F. Zheng, X. Guo, F. Huang, X. Sun, W. Zheng, Viewpoint-aware loss with angular regularization for person re-identification, in: AAAI, 2020, pp. 13114–13121. doi:10.1609/aaai.v34i07.7014.

[21] G. Wang, J. Lai, P. Huang, X. Xie, Spatial-temporal person re-identification, in: AAAI, 2019, pp. 8933–8940. doi:10.1609/aaai.v33i01.33018933.

[22] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, Y. Yang, Improving person re-identification by attribute and identity learning, Pattern Recognition 95 (2019) 151–161. doi:10.1016/j.patcog.2019.06.006.

[23] Z. Zhang, C. Lan, W. Zeng, Z. Chen, Densely semantically aligned person re-identification, in: CVPR, 2019, pp. 667–676. doi:10.1109/CVPR.2019.00076.

[24] X. Jin, C. Lan, W. Zeng, G. Wei, Z. Chen, Semantics-aligned representation learning for person re-identification, in: AAAI, 2020, pp. 11173–11180. doi:10.1609/aaai.v34i07.6775.

[25] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, Communications of the ACM 60 (6) (2017) 84–90. doi:10.1145/3065386.

[26] Z. Dai, M. Chen, X. Gu, S. Zhu, P. Tan, Batch dropblock network for person re-identification and beyond, in: ICCV, 2019, pp. 3690–3700. doi:10.1109/ICCV.2019.00379.

[27] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, C. Bregler, Efficient object localization using convolutional networks, in: CVPR, 2015, pp. 648–656. doi:10.1109/CVPR.2015.7298664.

[28] X. Chen, C. Fu, Y. Zhao, F. Zheng, J. Song, R. Ji, Y. Yang, Salience-guided cascaded suppression network for person re-identification, in: CVPR, 2020, pp. 3297–3307. doi:10.1109/CVPR42600.2020.00336.

[29] W. Li, X. Zhu, S. Gong, Harmonious attention network for person re-identification, in: CVPR, 2018, pp. 2285–2294. doi:10.1109/CVPR.2018.00243.

[30] J. Sun, Y. Li, H. Chen, B. Zhang, J. Zhu, MEMF: Multi-level-attention embedding and multi-layer-feature fusion model for person re-identification, Pattern Recognition 116 (2021) 107937. doi:10.1016/j.patcog.2021.107937.

[31] T. Si, F. He, H. Wu, Y. Duan, Spatial-driven features based on image dependencies for person re-identification, Pattern Recognition 124 (2022) 108462. doi:10.1016/j.patcog.2021.108462.

[32] Y. Chen, H. Wang, X. Sun, B. Fan, C. Tang, H. Zeng, Deep attention aware feature learning for person re-identification, Pattern Recognition 126 (2022) 108567. doi:10.1016/j.patcog.2022.108567.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: CVPR, 2016, pp. 2818–2826. doi:10.1109/CVPR.2016.308.

[34] X. Wang, X. Han, W. Huang, D. Dong, M. R. Scott, Multi-similarity loss with general pair weighting for deep metric learning, in: CVPR, 2019, pp. 5022–5030. doi:10.1109/CVPR.2019.00516.

[35] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, Q. Tian, Scalable person re-identification: A benchmark, in: ICCV, 2015, pp. 1116–1124. doi:10.1109/ICCV.2015.133.

[36] Z. Zhong, L. Zheng, D. Cao, S. Li, Re-ranking person re-identification with k-reciprocal encoding, in: CVPR, 2017. doi:10.1109/cvpr.2017.389.

[37] L. Wei, S. Zhang, W. Gao, Q. Tian, Person transfer gan to bridge domain gap for person re-identification, in: CVPR, 2018, pp. 79–88. doi:10.1109/CVPR.2018.00016.

[38] W. Li, R. Zhao, T. Xiao, X. Wang, Deepreid: Deep filter pairing neural network for person re-identification, in: CVPR, 2014, pp. 152–159. doi:10.1109/CVPR.2014.27.

[39] D. Fu, D. Chen, J. Bao, H. Yang, L. Yuan, L. Zhang, H. Li, D. Chen, Unsupervised pre-training for person re-identification, in: CVPR, 2021. doi:10.1109/cvpr46437.2021.01451.

[40] X. Fan, W. Jiang, H. Luo, M. Fei, Spherereid: Deep hypersphere manifold embedding for person re-identification, Journal of Visual Communication and Image Representation 60 (2019) 51–58. doi:10.1016/j.jvcir.2019.01.010.

[41] H. O. Song, Y. Xiang, S. Jegelka, S. Savarese, Deep metric learning via lifted structured feature embedding, in: CVPR, 2016. doi:10.1109/cvpr.2016.434.

[42] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, Y.-D. Shen, Dual-path convolutional image-text embeddings with instance loss, ACM Transactions on Multimedia Computing, Communications, and Applications 16 (2) (2020) 1–23. doi:10.1145/3383184.

[43] Z. Zheng, L. Zheng, Y. Yang, A discriminatively learned CNN embedding for person reidentification, ACM Transactions on Multimedia Computing, Communications, and Applications 14 (1) (2018) 1–20. doi:10.1145/3159171.

[44] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, Y. Wei, Circle loss: A unified perspective of pair similarity optimization, in: CVPR, 2020. doi:10.1109/cvpr42600.2020.00643.

[45] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, SphereFace: Deep hypersphere embedding for face recognition, in: CVPR, 2017. doi:10.1109/cvpr.2017.713.

[46] X. Liu, W. Liu, T. Mei, H. Ma, A deep learning-based approach to progressive vehicle re-identification for urban surveillance, in: European conference on computer vision, 2016, pp. 869–884. doi:10.1007/978-3-319-46475-6_53.

[47] R. Quan, X. Dong, Y. Wu, L. Zhu, Y. Yang, Auto-reid: Searching for a part-aware convnet for person re-identification, in: ICCV, 2019, pp. 3749–3758. doi:10.1109/ICCV.2019.00385.

[48] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: CVPR, 2019, pp. 5693–5703. doi:10.1109/CVPR.2019.00584.

[49] H. Gu, J. Li, G. Fu, M. Yue, J. Zhu, Loss function search for person re-identification, Pattern Recognition 124 (2022) 108432. doi:10.1016/j.patcog.2021.108432.

[50] K. Zhou, Y. Yang, A. Cavallaro, T. Xiang, Omni-scale feature learning for person re-identification, in: ICCV, 2019, pp. 3701–3711. doi:10.1109/ICCV.2019.00380.

[51] B. Bryan, Y. Gong, Y. Zhang, C. Poellabauer, Second-order non-local attention networks for person re-identification, in: ICCV, 2019, pp. 3759–3768. doi:10.1109/ICCV.2019.00386.

[52] G. Chen, C. Lin, L. Ren, J. Lu, J. Zhou, Self-critical attention learning for person re-identification, in: ICCV, 2019, pp. 9636–9645. doi:10.1109/ICCV.2019.00973.