

# AutoLoss-GMS: Searching Generalized Margin-based Softmax Loss Function for Person Re-identification

Hongyang Gu<sup>1,2</sup> Jianmin Li<sup>2\*</sup> Guangyuan Fu<sup>1</sup> Chifong Wong<sup>2</sup> Xinghao Chen<sup>3</sup> Jun Zhu<sup>2</sup>

<sup>1</sup>Xi'an Research Institute of High Technology, Xi'an, China

<sup>2</sup>State Key Laboratory of Intelligent Technology and Systems, BNRist, Institute for AI, Department of Computer Science and Technology, Tsinghua University, Beijing, China

<sup>3</sup>Huawei Noah's Ark Lab, Beijing, China

guhy18@mails.tsinghua.edu.cn lijianmin@mails.tsinghua.edu.cn fugy402@gmail.com

hzf19@mails.tsinghua.edu.cn xinghao.chen@outlook.com dcszj@mails.tsinghua.edu.cn

## Abstract

*Person re-identification is a hot topic in computer vision, and the loss function plays a vital role in improving the discrimination of the learned features. However, most existing models utilize the hand-crafted loss functions, which are usually sub-optimal and challenging to be designed. In this paper, we propose a novel method, AutoLoss-GMS, to search the better loss function in the space of generalized margin-based softmax loss function for person re-identification automatically. Specifically, the generalized margin-based softmax loss function is first decomposed into two computational graphs and a constant. Then a general searching framework built upon the evolutionary algorithm is proposed to search for the loss function efficiently. The computational graph is constructed with a forward method, which can construct much richer loss function forms than the backward method used in existing works. In addition to the basic in-graph mutation operations, the cross-graph mutation operation is designed to further improve the offspring's diversity. The loss-rejection protocol, equivalence-check strategy and the predictor-based promising-loss chooser are developed to improve the search efficiency. Finally, experimental results demonstrate that the searched loss functions can achieve state-of-the-art performance and be transferable across different models and datasets in person re-identification.*

## 1. Introduction

Person re-identification (ReID) [1–3] aims at retrieving an interested person across multiple, non-overlapping cameras. With the advancement of deep neural networks

(DNNs) and the increasing demand for intelligent video surveillance, ReID attracts more and more attention from the computer vision community. Although DNN-based models have made a significant breakthrough in ReID, learning discriminative features to identify the person from the large-scale gallery set is still challenging due to significant intra-class variance caused by pose variations, occlusions, or cluttered backgrounds.

Recently, most works [2, 4–7] have been based on the network design to obtain the discriminative features and ignore the importance of the loss function. As we all know, a high-performance DNN model is inseparable from the well-designed network architecture and the appropriate loss function. However, most existing works still adopt the paradigm of cross-entropy loss and triplet loss, and there are very few works exploring other forms of loss function on the ReID problem. Inspired by NormFace [8] in face recognition (FR), Fan *et al.* [3] proposed SphereReID to learn a hypersphere manifold embedding, which is better at extracting the discriminative features than the cross-entropy loss function. But according to recent researches in FR, NormFace is not the best choice among the margin-based softmax (MS) loss functions [9–11]. Sun *et al.* [12] proposed a more flexible loss function named CircleLoss, which is superior to the other MS loss functions in FR and ReID tasks. The loss functions mentioned above are all special cases in the generalized margin-based softmax (GMS) loss function and just the tip of the iceberg. Likely, they are not optimal in this space. Therefore, we aim to explore whether there is a better loss function than these classical loss functions in the GMS loss function space.

With the development of AutoML, more and more automated methods are proposed. Especially in the fields of data augmentation [13–16] and networks architecture [17–19], automated methods have surpassed hand-crafted methods.

\*Corresponding author.

In the field of the loss function, the AutoML-based methods have also emerged recently. Existing works mainly use two methods to search the loss function: 1) The first method does not directly search for the specific form of the loss function [20, 21], but, like PBT [22], only cares about the final trained neural network. Therefore, once the dataset or network changes, the expensive search procedure will be implemented again. 2) The second method can search for the loss function composed of some predefined primitive operations [23, 24] or the metric-surrogate loss with parameterized function [25]. What's more, the searched loss in the second method can be directly transferred to other similar tasks without searching again. However, the second searching method often requires a lot of search cost like other AutoML methods, so it is critical to improve the search efficiency. Although many methods [23, 24] have been proposed to speed up the search, the improvement of search efficiency is still limited.

In this paper, a novel method named AutoLoss-GMS is proposed to automatically search the loss functions in the GMS loss function space for person re-identification based on the second searching method. Specifically, we first represent the GMS loss function with two computational graphs and a constant. Then a general searching framework built upon the evolutionary algorithm is proposed to efficiently search for the loss function. The computational graph is constructed with a forward method, which can construct much richer loss function forms than the backward method used in [23, 24]. In addition to the basic in-graph mutation operations, the cross-graph mutation operation is designed to further improve the offspring's diversity. The loss-rejection protocol, equivalence-check strategy and the predictor-based promising-loss chooser are developed to improve the search efficiency. To the best of our knowledge, AutoLoss-GMS is the first work to use predictor in loss function search (LFS). We summarize the contributions of this work as follows:

- AutoLoss-GMS is the first work to search the GMS loss function for person re-identification.
- A general searching framework with the performance predictor is proposed to efficiently search the GMS loss function.
- The searched loss functions are transferable across different models and datasets with competitive performance and can achieve state-of-the-art performance in person re-identification.

## 2. Related Works

### 2.1. Person Re-identification

In recent years, person ReID has become a hot topic in the computer vision community. With the support of DNN, the performance of DNN-based models has surpassed the

level of humans. The key to the ReID problem is to extract discriminative features. Most of the existing works start with the network architecture [4, 26, 27] design to improve the discrimination of features. However, the loss function also plays a vital role in extracting discriminative features. In ReID, cross-entropy and triplet are commonly used paradigms, and there are very few works exploring other forms of loss functions. Inspired by NormFace [8] in FR, Fan *et al.* [3] proposed SphereReID, which experimentally verified that the features embedded on the hypersphere are more discriminative than the features learned by cross-entropy. However, SphereReID only considers the hyperspherical embedding instead of the large margin, which has been verified in FR to perform better than NormFace [9–11]. Sun *et al.* [12] proposed a more flexible loss named CircleLoss, which significantly improved compared with other MS loss functions. However, designing such a good loss function requires a lot of professional knowledge and energy for humans. Therefore, AM-LFS [20] tries to automatically design the loss function in the MS space to reduce the burden on humans. However, AM-LFS cannot get a searched loss function with a fixed form, which is not conducive to directly transferring to other datasets or networks without searching again.

### 2.2. Loss Function Search

The loss function is an indispensable part of deep learning, but the hand-crafted loss function often requires much professional knowledge to design. With the development of AutoML, the LFS has gradually become the pursued goal. AM-LFS [20] and Searched-Softmax [28] are two similar works, and they both use reinforcement learning to search  $t(x)$  under the framework of the MS loss. However, the searched losses of these two methods do not have a fixed form, so they are hard to directly transfer across datasets and networks without searching again. Some other methods are closer to the network architecture search (NAS), and a fixed form of result can be obtained. Auto Seg-Loss [25] searches for the differentiable part to replace the non-differentiable part of the metrics, but this method is difficult to apply to our search space. AutoLoss-Zero [24] and CSE-AutoLoss [23] use a computational graph composed of primitive mathematical operations to represent the loss function. This loss function representation method meets our goal, but this searching method requires a large amount of search cost. Therefore AutoLoss-Zero and CSE-AutoLoss propose various methods to improve search efficiency, but the improvement is still limited.

## 3. Method

In this section we provide the detailed introduction to our proposed method, including the design of search space and the search algorithm.

Table 1. The  $t(x)$  and  $n(x)$  of the hand-crafted loss functions. The  $m$  is the hyper-parameter in the corresponding loss function. The  $\text{de}(x)$  represents the “Detach” operation.

Loss	$t(x)$	$n(x)$
NormFace	$x$	$x$
CosFace	$x - m$	$x$
ArcFace	$\cos(\arccos(x) + m)$	$x$
SphereFace	$\cos(m \arccos(x))$	$x$
CircleLoss	$[\text{de}(1 + m - x)]_+ (x - 1 + m)$	$[\text{de}(m + x)]_+ (x - m)$

### 3.1. Search Space

#### 3.1.1 Preliminary Knowledge

To have a clear understanding of the search space, some preliminary knowledge is introduced first. To start with, we consider the standard softmax cross-entropy loss:

$$L_{ce} = -\log \frac{\exp(\mathbf{W}_y^T \mathbf{x} + b_y)}{\sum_{i=1}^K \exp(\mathbf{W}_i^T \mathbf{x} + b_i)}; \quad (1)$$

where  $\mathbf{x} \in \mathbb{R}^d$  denotes the input feature vector,  $y \in \{1, 2, \dots, K\}$ ;  $K$  is its ground truth label,  $K$  is the total number of classes,  $\mathbf{W}_i \in \mathbb{R}^d$  and  $b_i \in \mathbb{R}$  are the weight vector and the bias of the  $i$ -th class, respectively. By removing the biases, normalizing the classifier weights and feature to one (i.e.,  $\|\mathbf{W}_i\|_2 = 1$ ;  $\|\mathbf{x}\|_2 = 1$ ), and adding the scale factor  $s > 0$ , the unified form of the MS loss function can be obtained:

$$L_{ms} = -\log \frac{\exp(s \cdot t(\cos y))}{\exp(s \cdot t(\cos y)) + \sum_{i \neq y} \exp(s \cdot \cos i)}; \quad (2)$$

where  $\cos i = (\mathbf{W}_i^T \mathbf{x}) / (\|\mathbf{W}_i\|_2 \|\mathbf{x}\|_2)$  and  $t(x)$  is a function whose domain is  $[-1, 1]$ . The GMS loss function can be obtained by further introducing  $n(x)$  in the non-target part in the MS loss function:

$$L_{gms} = -\log \frac{\exp(s \cdot t(\cos y))}{\exp(s \cdot t(\cos y)) + \sum_{i \neq y} \exp(s \cdot n(\cos i))}; \quad (3)$$

Obviously, the MS loss function is a special case of the GMS loss function, where  $n(x) = x$ . As shown in Table 1, the commonly used MS loss functions [8–11] and CircleLoss [12] can be represent by setting different  $t(x)$  and  $n(x)$ . In this paper, the form of the loss function we aim to explore is based on  $L_{gms}$ , that is, search for the specific expressions of  $t(x)$ ,  $n(x)$  and the specific value of  $s$ . The search target can be formulated as a nested optimization:

$$\begin{aligned} & \mathcal{M}^* = \arg \max_{\mathcal{M}} \mathcal{M}(\mathbf{x}; D_{val}); \\ \text{s.t.: } & \mathcal{M}(\mathbf{x}) = \arg \min_{\mathcal{M}} E_{(\mathbf{x}, y) \in D_{trn}} L_{gms}(\mathcal{M}(\mathbf{x}); y); \end{aligned} \quad (4)$$

where  $\mathcal{M}$  is a network parameterized by  $\theta$ , the training dataset and validation dataset are denoted as  $D_{trn}$  and  $D_{val}$ , respectively.  $\mathcal{M}$  is a given evaluation metric, and  $L_{gms}$  is  $L_{gms}$  parameterized by  $\theta = fs; t(x); n(x)g$ .

#### 3.1.2 Loss Function Representation

As mentioned in the previous section, our target is to search for the optimal  $\theta = fs; t(x); n(x)g$ . Since the roles of  $t(x)$ ,  $n(x)$  and  $s$  in  $L_{gms}$  are decoupled, we can consider the representation of these three parts separately.

We define the scale factor  $s > 1$ , which is consistent with all hand-crafted loss functions. We discretize  $s$  as

$$s \in Scales = \{2^{-s} | j \in \mathbb{N}; 0 \leq j \leq N_s\}; \quad (5)$$

where  $s > 0$  and  $N_s$  are the predefined values. Although this discretization is simple, it is sufficient to achieve good performance in the subsequent experiments.

For  $t(x)$  and  $n(x)$ , we only introduce the representation of  $t(x)$ , and  $n(x)$  is similar. The  $t(x)$  can be represented as a computational graph (CG)  $G_t$ :

- There are two types of input nodes in the  $G_t$  as shown the green and gray nodes in Figure 2. The green node  $x$  is the network output  $\cos y$  in  $L_{gms}$ . The gray nodes are the predefined constants, which are represented like the scale factor  $s$  and defined as

$$c \in Cons = \{f_c | j \in \mathbb{N}; 0 \leq j \leq N_c\}; \quad (6)$$

where  $c > 0$  and  $N_c$  are the predefined values.

- The intermediate computational nodes (the blue nodes in Figure 2) are the primitive mathematical operations selected from the set  $H$  as shown in Table 1 of Suppl. A.
- The result of the output node (the orange node  $O$  in Figure 2) is the  $t(x)$ .

To sum up, the  $\theta = fs; t(x); n(x)g$  is represented as  $s \in Scales$ ,  $G_t$  and  $G_n$ , which are our search space.

#### 3.2. Search Algorithm

Our search algorithm is mainly based on the variants of evolutionary algorithm [24, 29, 30], which are easy to be parallelized by distributing training. The search pipeline of our algorithm is illustrated in Figure 1. Firstly,  $K$  loss functions are generated to form the initial population. Then in each evolution, two parent loss functions are selected by two independent tournament selections [31] ( $T$  ratio of current population). These two selected loss functions are used to produce the offspring through the well-designed mutation operations until the offspring pass the loss-rejection protocol. The equivalence-check strategy based on the feature vector is proposed to avoid accurately re-evaluating the mathematically equivalent loss functions. The most promising loss function would be chosen among the non-equivalent loss functions by the predictor-based promising-loss chooser. Following [24, 30], only the most recent  $P$  loss functions are maintained in the population.

Figure 1. The search pipeline of AutoLoss-GMS.

### 3.2.1 Initialization and Mutation

We apply two methods to initialize the population: 1) **randomly-generated loss functions** for searching from scratch like AutoLoss-Zero [24] and 2) **predefined hand-crafted loss functions** for searching from prior knowledge like CSE-Autoloss [23]. The predefined hand-crafted loss functions are mainly based on Table 1.

AutoLoss-Zero and CSE-Autoloss construct the CGs in a backward manner [23, 24], but this construction method is not efficient. Specifically, the output of the intermediate node cannot be used multiple times, as shown in the purple part of Figure 2(b). To construct CGs more efficiently, a forward construction method is proposed inspired by the network architecture search (NAS) [17]. The graph constructed by forward method starts with  $1 + n_c$  input nodes, which are one feature node  $x$  (the green node in Figure 2(a)) and  $n_c$  constant nodes (the gray nodes in Figure 2(a)). Then, given the state of the current CG, all possible operations and edge connections are enumerated, and a possible operation and its corresponding edge connections are randomly selected to add to the CG. When the selected operation is the “Output” operation, all nodes without successors are connected to the output node  $O$ . When the CG reaches the maximum number of primitive operations  $n_o$ , only the “Output” operation can be chosen. Under the same number of primitive operations, the forward method can construct much richer loss function forms than the backward method. The forward construction process of a CG is shown in Algorithm 1 of Suppl. B, and the loss function of our search space can be constructed by Algorithm 2 in Suppl. B.

In addition to the three in-graph mutation operations (*In-*

*sertion*, *Deletion* and *Replacement* are detailed in Suppl. C) like AutoLoss-Zero [24], a cross-graph mutation operation is introduced. Specifically, in each evolution, given two loss functions  $(^{(1)})$  and  $(^{(2)})$ ,  $G_t$  or  $G_n$  in  $(^{(2)})$  replaces the corresponding CG in  $(^{(1)})$  with the probability of  $p_c$  to obtain the mutated  $^{(j)}$ , and the  $s^{(j)}$  in  $^{(j)}$  is changed as follows:

$$s^{(j)} = 2^{[(\log_2(s^{(1)}) + \log_2(s^{(2)})) \div 2] - s^{(j)}}, \quad (7)$$

where  $[x]_s$  means standardizing the precision of  $x$  to  $s$ . Generally speaking, the hyper-parameters in the loss function will have a significant impact on the results, so we also introduce the mutation operations on the  $s$  and  $fcon_{i,j=1}^{n_c} g$ , instead of keeping these constants fixed like AutoLoss-Zero and CSE-Autoloss. The specific mutation operations for the constant term are as follows:

$$\begin{aligned} s^{(j)} &= 2^{\text{Clamp}[\log_2(s) + s \cdot r; 0; s \cdot N_s]} \\ c^{(j)} &= \text{Clamp}[c + c \cdot r; 0; c \cdot N_c] \end{aligned} \quad (8)$$

where  $\text{Clamp}[x; x_{\min}; x_{\max}]$  means clamp  $x$  in  $[x_{\min}; x_{\max}]$ , and  $r$  is randomly selected from  $\mathcal{F}[1; 0; 1]g$ .

Given the two loss functions  $(^{(1)})$  and  $(^{(2)})$  selected by tournament, the offspring is produced by Algorithm 3 in Suppl. C.

### 3.2.2 Loss-Rejection Protocol

Inspired by CES-Autoloss [23] and AutoLoss-Zero [24], our loss-rejection protocol starts from two aspects:

**Basic properties protocol.** By analyzing the GMS loss function  $L_{gms}$ , the following properties should be generally met (see the Suppl. D for details):

Figure 2. The  $G_t$  of  $t(x)$  in CircleLoss. There are two constant nodes (the gray nodes) in the graph, and  $m = 0.25$ . The graph in (a) is constructed by forward method, while the graph in (b) is constructed by backward method. The numbers in the red nodes represent the order of construction.

- $t^0(x) > 0; x \in [1; 1]$ ;
- $n^0(x) > 0; x \in [1; 1]$ ;
- $n(x) - t(x) > 0; x \in [1; 1]$ .

**Toy task.** Given  $B$  random samples  $f(\mathbf{x}_b; y_b)_{b=1}^B g$  from the training dataset  $D_{trn}$  and a randomly initialized network  $\mathcal{M}_0$ , we record the network predictions and the corresponding labels as  $f(\mathbf{x}_b) = \mathcal{M}_0(\mathbf{x}_b; y_b)_{b=1}^B g$ . The toy task is set up to solve the following problem by only optimizing on  $\mathbf{x}_b_{b=1}^B$ :

$$\mathbf{x}_b_{b=1}^B = \arg \min_{f(\mathbf{x}_b)_{b=1}^B g} \frac{1}{B} \sum_{b=1}^B L_{gms}(\mathbf{x}_b; y_b); \quad (9)$$

To sum up, the loss-rejection protocol first judges the basic properties of  $\mathcal{L}$  and then performs the toy task. Only the  $\mathcal{L}$  that satisfies the basic properties and the evaluation metric  $(\mathbf{x}_b)_{b=1}^B g$  on the toy task is greater than the predefined threshold  $\tau_{toy}$  can pass the protocol, otherwise it will be rejected.

### 3.2.3 Equivalence-Check Strategy

The AutoLoss-Zero use the initial gradient norm  $f @ L_{gms} = @ \mathbf{x}_b_{b=1}^B g$  in toy task to judge the mathematically equivalent loss functions, but this may be inaccurate when only using the initial gradient norm. Considering the relationship among  $s$ ,  $t(x)$  and  $n(x)$  in  $L_{gms}$ , the  $\mathcal{L}$  can be formalized as a concise feature vector, which can accurately judge the equivalent loss functions.

Firstly, we discrete  $\mathcal{L}$  as

$$d = [f_t; f_n; \log_2(s) = (\frac{1}{s} N_s)] \in \mathbb{R}^{2N+1}; \quad (10)$$

where  $f_t \in \mathbb{R}^N$  and  $f_n \in \mathbb{R}^N$  are the values of  $t(x)$  and  $n(x)$  at uniformly discrete points in the domain  $[1; 1]$ . But the  $d$  cannot recognize the mathematically equivalent loss functions under the translation-scale transformation, for example,  $\mathcal{L}_0 = f t(x); n(x); sg$  and  $\mathcal{L}_{k;b} = f t(x)=k+b; n(x)=k+b; ksg; 8k \neq 0; b \in \mathbb{R}$  are mathematically equivalent for  $L_{gms}$ . We introduce the following transformation to normalize  $t(x)$  and  $n(x)$ :

$$\begin{aligned} TN_{\min} &= \min_{x \in [1; 1]} \min_{x \in [1; 1]} t(x); \min_{x \in [1; 1]} n(x) \\ TN_{\max} &= \max_{x \in [1; 1]} \max_{x \in [1; 1]} t(x); \max_{x \in [1; 1]} n(x) \\ b &= (TN_{\min} + TN_{\max})/2 \\ k &= \max(f(TN_{\max} - TN_{\min})/2; 1)sg \\ s &= k/s \\ t(x) &= (t(x) - b)/k \\ n(x) &= (n(x) - b)/k \end{aligned} \quad ; \quad (11)$$

where the transformed  $\bar{s} > 1$ , and the range of  $\bar{t}(x)$  and  $\bar{n}(x)$  is in  $[1; 1]$ . The  $d$  can perfectly recognize the mathematically equivalent loss functions under the translation-scale transformation. But the  $d \in [2N] \times [0; +1]$ , which seriously affects the prediction performance of predictor proposed next section. A scale constraint on the search space is proposed to solve this problem. Specifically, the generated CG needs to meet the following constraint:

$$\log_2((TN_{\max} - TN_{\min})/s) \leq 6; \quad (12)$$

where  $\bar{s} \geq s N_s$ . Under such constraint, the value range of  $\log_2(\bar{s})$  becomes in  $[0; 6]$ . In summary, we define the feature vector of a loss function with a parameter of  $\mathcal{L}$  as:

$$fv = [f_t; f_n; 2 \log_2(\bar{s}) = \log_2(s N_s)]; \quad (13)$$

Each element in  $fv$  is in the interval of  $[1; 1]$ , which achieves the purpose of normalization for the predictor training. What's more, the  $fv$  can be used as a equivalence-check to check the mathematically equivalent loss functions, which avoids re-evaluating the equivalent loss functions.

### 3.2.4 Promising-Loss Chooser

Although the loss-rejection protocol can filter the invalid loss functions, some loss functions passing the protocol still perform poorly. In order to further save the computational budget, inspired by the performance predictor in NAS [32, 33], we propose a CNN-based loss function performance predictor to further choose the most promising loss from the candidate population.

When the number of loss functions evaluated on the proxy task reaches  $E_0$ , the predictor  $P$  will be trained on the



Table 2. The searched loss functions. PIP: predefined initial populations. Each loss is searched on a certain dataset using certain model with PIP or not.

Searched Loss	Model	Dataset	PIP	$t(x)$	$n(x)$	$\log_2(s)$
AutoLoss-GMS-Zero	ResNet50	Market-1501	7	$(0.22 + e^{\frac{P}{0.22}})x + \arcsin(0.22)^2$	$x + 0.85$	5.5
AutoLoss-GMS-A	ResNet50	Market-1501	3	$\text{de}(1-x)(x-0.7)$	$[\text{Gd}(x \cdot \text{Sig}(0.25))]_+$	7.5
AutoLoss-GMS-B	ResNet50	CUHK03	3	$\text{de}(1.3-x)(x-1.0)$	$0.35x - 0.35^2$	6.0
AutoLoss-GMS-C	OSNet	Market-1501	3	$(x-0.84)(0.95-x)$	$[\text{de}(\arcsin(x))]_+ (x-0.5) + 0.05$	7.5
AutoLoss-GMS-D	MGN	Market-1501	3	$x + 0.15$	$x + 0.2$	4.0

current evaluated set (loss function  $f_i$  and the corresponding performance  $p_i$  pair)  $Eva = f(x_i; p_i) \prod_{i=1}^{E_0} g$ . Then every time the number of  $jEva_j$  increases by  $E$ , the predictor  $P$  is updated once according to the current  $Eva$ . After the predictor is trained for the first time, each produced new loss function that passes the equivalence-check strategy will be added to the candidate population of the promising-loss chooser. When the number of candidate population reaches the predefined  $N_p$ , the most promising one is chosen according to the results predicted by the current predictor, and the candidate population is cleared. The algorithm of choosing the promising loss is shown in Suppl. E Algorithm 4.

In this paper, we choose ResNet [34] as the CNN-based predictor, some adjusted details are shown in Suppl. F. Formally, given  $B$  different converted loss functions and their ground-truth performance  $f(x_i; y_i) \prod_{i=1}^B g$ , and  $fP(x_i) \prod_{i=1}^B g$  is the output of the predictor. The common used MSE loss function can be defined as:

$$L_{MSE} = \frac{1}{B} \sum_{i=1}^B (P(x_i) - y_i)^2; \quad (14)$$

As [32, 33] have pointed out, the predictor to determines the ranking of loss functions is more robust than the predictor to accurately predict the performance of loss functions. The Kendall's Tau (KTau) [35] is an indicator to measure the ranking relationship:

$$KTau = \frac{1}{C_B^2} \sum_{1 \leq i < j \leq B} \text{sign}(y_i - y_j) \text{sign}(P(x_i) - P(x_j)); \quad (15)$$

where  $\text{sign}(x)$  is the sign function. Inspired by [36], we convert the non-differentiable KTau to the differentiable KTau by replacing the second  $\text{sign}(x)$  in Eq.(15) with  $\tanh(x)$ :

$$L_K = \frac{1}{C_B^2} \sum_{1 \leq i < j \leq B} \text{sign}(y_i - y_j) \tanh(P(x_i) - P(x_j)); \quad (16)$$

where  $\beta$  governs the temperature of the  $\tanh(x)$  that replaces the  $\text{sign}(x)$  function. Therefore, the final loss function for our predictor is:

$$L = L_{MSE} + \beta L_K; \quad (17)$$

where  $\beta$  is the hyper-parameter that controls the importance between two different loss functions.

Figure 3. The analysis of the predictor. PLC: promising-loss chooser.

## 4. Experiments

### 4.1. Datasets and Evaluation Metrics

To verify the performance of AutoLoss-GMS, we conduct experiments on three common used datasets: Market-1501 [37], CUHK03 [38] and MSMT17 [39].

Following conventions in the ReID community [37, 39], all methods are evaluated with Cumulative Matching Characteristic (CMC) curves and the mean Average Precision (mAP).

### 4.2. Implementation Details

For the search algorithm, the population is initialized with  $K = 20$  loss functions and is restricted to the most recent  $P = 1000$  loss functions. The ratio of tournament selection is set as  $T = 5\%$  of the current population. The condition for the search to stop is evaluating 500 models on the proxy task. The search process takes around 2.5 days on five NVIDIA Telsa-V100 GPUs. More details are in Suppl. G. Five loss functions are obtained under different settings, as shown in Table 2.

### 4.3. Ablation Study

#### 4.3.1 Analysis of the predictor

We first randomly sample 1000 loss functions, and their performance on the proxy task is used to verify the effectiveness of the components in the predictor. Under different settings, ten times of training is implemented. A certain amount of data is randomly selected as the training dataset

Table 3. The effectiveness of the components in AutoLoss-GMS when searching on Market-1501 with ResNet50. RS: random search. EA: evolutionary algorithm. CGM: cross-graph mutation. LRP: loss-rejection protocol. BPP: basic properties protocol. TT: toy task. ECS: equivalence-check strategy. PIP: predefined initial population. PLC: promising-loss chooser.

		mAP	Speed-Up	Explored Losses
RS	Backward	81.32	1	500
	Forward	82.67	1	500
<hr/>				
	Naïve EA	83.39	1	500
	+CGM	83.77	1	500
	+LRP(BPP)	84.52	30	1.5 $10^4$
	+LRP(TT)	84.95	36	1.8 $10^4$
	+ECS	85.24	54	2.7 $10^4$
	+PIP	85.80	58	2.9 $10^4$
	+PLC	<b>88.02</b>	<b>540</b>	<b>2.7 <math>10^5</math></b>

in each training, and the rest is used as the test dataset. The training results are shown in Figure 3(a). The following conclusions can be drawn from the experimental results: 1) The result of simply using  $L_{MSE}$  is worse than  $L_{MSE} + L_K$ , which shows that our differentiable K $\tau$  is effective; 2) The unnormalized  $d$  performs poorly; 3) Converting  $f_V$  into a three-channel  $x$  performs well.

In Figure 3(b), we show the performance of AutoLoss-GMS with or without predictors. It can be seen from the results that when the predictor is introduced after searching 100 loss functions, the loss functions with poor performance are filtered out a lot, which further illustrates the effectiveness of our promising-loss chooser.

#### 4.3.2 Effectiveness of the components in AutoLoss-GMS

To verify the effectiveness of the components in our AutoLoss-GMS, the searching with different components are implemented, and the results are shown in Table 3. The following conclusions can be concluded from the results: 1) The forward construction is better than the backward construction, which verifies that the CGs constructed by the forward method have richer loss function forms; 2) The proposed cross-graph mutation operation can improve the performance in search; 3) The two-step loss-rejection protocol can expand the amount of loss function explored by 36 times, and quickly filters out the loss functions that cannot meet the basic properties or perform poorly on the toy task; 4) 50% more loss functions can be explored further by avoiding the re-evaluation on equivalent loss functions with the equivalence-check strategy; 5) The prior knowledge introduced by the hand-crafted loss function is conducive to better search results; 6) The predictor-based promising-loss chooser can further improve search efficiency by about ten times.

Table 4. Transferability of the searched losses and hand-crafted losses. Only the mAP metric is reported. R: ResNet50. O: OS-Net. The results in the bracket represent the results after directly being transferred across datasets or networks without any modification on hyper-parameters. The results outside the bracket represent the results after fine-tuning  $s$  for our searched loss or other hyper-parameters for hand-crafted losses. When there is only one result, it is the performance of the fine-tuned hand-crafted losses or the loss searched under this setting.

	Methods	Market-1501	CUHK03
R	NormFace	81.74	58.48(55.91)
	CosFace	84.23	61.94(57.41)
	ArcFace	85.06	65.03(56.54)
	CircleLoss	85.41	66.87(59.21)
	AutoLoss-GMS-A	<b>87.00</b>	<b>68.33(67.18)</b>
	AutoLoss-GMS-B	78.03(76.10)	67.11
	AutoLoss-GMS-C	86.64(86.23)	67.95(67.79)
	AutoLoss-GMS-Zero	84.16	64.32(64.32)
O	AutoLoss-GMS-A	<b>88.99(88.87)</b>	70.60(70.60)
	AutoLoss-GMS-B	86.20(84.65)	71.15(70.27)
	AutoLoss-GMS-C	88.94	<b>72.57(72.17)</b>
	AutoLoss-GMS-Zero	85.79(84.59)	68.95(67.59)

#### 4.3.3 Transferability of the searched losses

To verify the transferability of the loss functions searched on different datasets and different networks, we conduct an in-depth comparison of these loss functions, as shown in Table 4. The hyper-parameters in the hand-crafted loss function are carefully fine-tuned on the two datasets by grid search shown in Figure 3 of Suppl. H. The mAP of our searched AutoLoss-GMS-A is 1.59% higher than the best CircleLoss among the hand-crafted loss functions. Without the predefined initial population, AutoLoss-GMS-Zero can achieve performance equivalent to CosFace. Maintaining the hyper-parameter settings on Market-1501, AutoLoss-GMS-A can still surpass CircleLoss' 59.21% with 67.18% mAP on the CUHK03 dataset. When the hyper-parameters on the CUHK03 dataset are further fine-tuned, the CircleLoss can reach 66.87% performance. To fast adapt to other datasets (networks), we can also perform a fine-tuning on the  $s$  of the searched loss. AutoLoss-GMS-A fine-tuned on CUHK03 can reach 68.33%, which is still 1.46% higher than CircleLoss.

AutoLoss-GMS-B and AutoLoss-GMS-C are used to test the performance across datasets and networks with AutoLoss-GMS-A. AutoLoss-GMS-B only performs well on CUHK03 but not good enough on the Market-1501. The main reason is that CUHK03 has fewer data than Market-1501. Therefore, searching on a large dataset can obtain a loss function with good generalization. AutoLoss-GMS-A and AutoLoss-GMS-C are obtained by searching on different models. Regardless of the fine-tuning on  $s$ , the performance of these two loss functions is almost the best under

Table 5. Comparison with state-of-the-art methods. The part above the double line is the models based on the global feature, and the part below the double line is the models based on the local feature. In each part, the ones below the single line are our methods, and the ones above are other methods. **Redbold** is the best performance of ours. **Bluebold** is the best performance of others.

Methods	Market-1501		CUHK03		MSMT17	
	mAP	Rank-1	mAP	Rank-1	mAP	Rank-1
TriNet [40]	69.1	84.9	-	-	-	-
SphereReID [3]	83.6	94.4	-	-	-	-
StrongBaseline [41]	85.9	94.5	-	-	-	-
ResNet50+CircleLoss [12]	84.9	94.2	-	-	50.2	76.3
OSNet [42]	<b>86.7</b>	94.8	67.8	72.3	<b>55.1</b>	<b>79.1</b>
AutoReID [43]	85.1	94.5	<b>69.3</b>	<b>73.3</b>	52.5	78.2
SphereReID + AM-LFS [20]	84.4	<b>95.0</b>	-	-	-	-
<b>ResNet50 + AutoLoss-GMS-A(ours)</b>	87.0	94.7	68.3	70.4	55.1	79.5
<b>OSNet + AutoLoss-GMS-C(ours)</b>	<b>88.9</b>	<b>95.7</b>	<b>72.6</b>	<b>74.5</b>	<b>62.6</b>	<b>83.7</b>
PCB+RPP [27]	81.6	93.8	57.5	63.7	-	-
MGN [4]	86.9	95.7	<b>66.0</b>	<b>66.8</b>	-	-
MGN+CircleLoss [12]	87.4	<b>96.1</b>	-	-	<b>52.1</b>	<b>76.9</b>
MGN + AM-LFS [20]	<b>88.1</b>	95.8	-	-	-	-
<b>MGN + AutoLoss-GMS-A(ours)</b>	88.7	95.6	73.2	75.2	58.2	80.9
<b>MGN + AutoLoss-GMS-C(ours)</b>	89.3	95.7	72.3	75.3	58.3	79.9
<b>MGN + AutoLoss-GMS-D(ours)</b>	<b>90.1</b>	<b>96.2</b>	<b>74.3</b>	<b>75.6</b>	<b>63.0</b>	<b>83.7</b>

their respective models. Even after transferring, the performance of the searched loss functions is enough to surpass the hand-crafted loss functions, which shows that our searched loss functions have a certain degree of transferability.

#### 4.4. Comparison With State-of-the-Art Methods

Our comparison with the SOTA methods is based on two types of models, namely, the models based on the global feature and the models based on the local feature. We report the performance of AutoLoss-GMS-A, AutoLoss-GMS-C and AutoLoss-GMS-D, and the comparison results are shown in Table 5.

Among the models based on the global feature with the ResNet50 backbone, ResNet50 + AutoLoss-GMS-A achieves the highest mAP on all three datasets. Especially on Market-1501, our AutoLoss-GMS-A can surpass StrongBaseline, which uses cross-entropy and triplet paradigms, by 1.1%. On the largest dataset, MSMT17, our AutoLoss-GMS-A significantly surpasses CircleLoss by 4.9% in terms of mAP. On all three datasets, under the support of AutoLoss-GMS-A, ResNet50 can achieve similar performance with the network architecture specially designed for ReID (*i.e.* OSNet, AutoReID). When equipped with OSNet, AutoLoss-GMS-C can further improve the original OSNet significantly. Especially on MSMT17, OSNet + AutoLoss-GMS-C can reach 62.6% mAP.

The performance of directly applying AutoLoss-GMS-A and AutoLoss-GMS-C to the MGN model is significantly better than MGN + CircleLoss, which further verifies that our searched loss functions are transferable across models. Furthermore, MGN + AutoLoss-GMS-D surpasses other methods on Market-1501 with an mAP of 90.1%.

In summary, the performance of our searched loss functions is significantly better than the performance of the hand-crafted CircleLoss and AutoML-based AM-LFS, which further illustrates the effectiveness of our method.

## 5. Conclusions

In this paper, we propose a novel method, AutoLoss-GMS, to search the better loss function in the space of generalized margin-based softmax loss function for person re-identification automatically. The experimental results demonstrate that the searched loss functions achieve state-of-the-art performance and are transferable across different models and datasets in person ReID.

However, current good search results still heavily depend on prior knowledge, and subsequent research will focus on further improving the performance of search results without prior knowledge. We will also plan to apply the searching framework on more tasks to further verify the effectiveness of our search algorithm.

**Acknowledgments** This work was supported by the National Key Research and Development Program of China (No. 2017YFA0700904), the National Natural Science Foundation of China (No. U1811461), and the Beijing NSF Project (No. JQ19016).



## References

- [1] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1
- [2] Guangyi Chen, Tianpei Gu, Jiwen Lu, Jin-An Bao, and Jie Zhou. Person re-identification via attention pyramid. *IEEE Transactions on Image Processing*, 2021. 1
- [3] Xing Fan, Wei Jiang, Hao Luo, and Mengjuan Fei. Sphered: Deep hypersphere manifold embedding for person re-identification. *Journal of Visual Communication and Image Representation*, 60:51–58, 2019. 1, 2, 8
- [4] Guanshuo Wang, Yufeng Yuan, Xiong Chen, Jiwei Li, and Xi Zhou. Learning discriminative features with multiple granularities for person re-identification. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 274–282, 2018. 1, 2, 8
- [5] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8522, 2019. 1
- [6] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *CoRR*, abs/2102.04378, 2021. 1
- [7] Hongyang Gu, Guangyuan Fu, Jianmin Li, and Jun Zhu. Auto-reid+: Searching for a multi-branch convnet for person re-identification. *Neurocomputing*, 435:53–66, 2021. 1
- [8] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface:  $L_2$  hypersphere embedding for face verification. In Qiong Liu, Rainer Lienhart, Haohong Wang, Sheng-Wei “Kuan-Ta” Chen, Susanne Boll, Yi-Ping Phoebe Chen, Gerald Friedland, Jia Li, and Shuicheng Yan, editors, *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, pages 1041–1049. ACM, 2017. 1, 2, 3
- [9] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 1, 2, 3
- [10] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 1, 2, 3
- [11] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 2, 3
- [12] Yifan Sun, Changmao Cheng, Yuhang Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. 1, 2, 3, 8
- [13] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 1
- [14] Sungbin Lim, Ildoo Kim, Taesup Kim, Chihyeon Kim, and Sungwoong Kim. Fast autoaugment. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6662–6672, 2019. 1
- [15] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 1
- [16] Keyu Tian, Chen Lin, Ming Sun, Luping Zhou, Junjie Yan, and Wanli Ouyang. Improving auto-augment via augmentation-wise weight sharing. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1
- [17] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 4
- [18] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 1
- [19] Xinbang Zhang, Zehao Huang, Naiyan Wang, Shiming Xiang, and Chunhong Pan. You only search once: Single shot neural architecture search via direct sparse optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2891–2904, 2021. 1
- [20] Chuming Li, Xin Yuan, Chen Lin, Minghao Guo, Wei Wu, Junjie Yan, and Wanli Ouyang. AM-LFS: automl for loss function search. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 8409–8418. IEEE, 2019. 2, 8
- [21] Xiaobo Wang, Shuo Wang, Cheng Chi, Shifeng Zhang, and Tao Mei. Loss function search for face recognition. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10029–10038. PMLR, 2020. 2

- [22] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *CoRR*, abs/1711.09846, 2017. 2
- [23] Peidong Liu, Gengwei Zhang, Bochao Wang, Hang Xu, Xiaodan Liang, Yong Jiang, and Zhenguo Li. Loss function discovery for object detection via convergence-simulation driven search. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2, 4
- [24] Hao Li, Tianwen Fu, Jifeng Dai, Hongsheng Li, Gao Huang, and Xizhou Zhu. Autoloss-zero: Searching loss functions from scratch for generic tasks. *CoRR*, abs/2103.14026, 2021. 2, 3, 4
- [25] Hao Li, Chenxin Tao, Xizhou Zhu, Xiaogang Wang, Gao Huang, and Jifeng Dai. Auto seg-loss: Searching metric surrogates for semantic segmentation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 2
- [26] Tianlong Chen, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren, and Zhangyang Wang. Abdn-net: Attentive but diverse person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8351–8361, 2019. 2
- [27] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 480–496, 2018. 2, 8
- [28] Xiaobo Wang, Shuo Wang, Cheng Chi, Shifeng Zhang, and Tao Mei. Loss function search for face recognition. In *International Conference on Machine Learning*, pages 10029–10038. PMLR, 2020. 2
- [29] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019. 3
- [30] Esteban Real, Chen Liang, David So, and Quoc Le. Automl-zero: Evolving machine learning algorithms from scratch. In *International Conference on Machine Learning*, pages 8007–8019. PMLR, 2020. 3
- [31] Seyedali Mirjalili. Genetic algorithm. In *Evolutionary algorithms and neural networks*, pages 43–55. Springer, 2019. 3
- [32] Lukasz Dudziak, Thomas C. P. Chau, Mohamed S. Abdelfattah, Royson Lee, Hyeji Kim, and Nicholas D. Lane. BRP-NAS: prediction-based NAS using gcns. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 5, 6
- [33] Yixing Xu, Yunhe Wang, Kai Han, Yehui Tang, Shangling Jui, Chunjing Xu, and Chang Xu. Renas: Relativistic evaluation of neural architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 4411–4420. Computer Vision Foundation / IEEE, 2021. 5, 6
- [34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6
- [35] Pranab Kumar Sen. Estimates of the regression coefficient based on kendall’s tau. *Journal of the American statistical association*, 63(324):1379–1389, 1968. 6
- [36] Andrew Brown, Weidi Xie, Vicky Kalogeiton, and Andrew Zisserman. Smooth-ap: Smoothing the path towards large-scale image retrieval. In *European Conference on Computer Vision*, pages 677–694. Springer, 2020. 6
- [37] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. 6
- [38] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*, pages 17–35. Springer, 2016. 6
- [39] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 79–88, 2018. 6
- [40] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 8
- [41] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 8
- [42] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3702–3712, 2019. 8
- [43] Ruijie Quan, Xuanyi Dong, Yu Wu, Linchao Zhu, and Yi Yang. Auto-reid: Searching for a part-aware convnet for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3750–3759, 2019. 8