Faster Person Re-Identification: One-shot-Filter and Coarse-to-Fine Search

Guan'an Wang*, Xiaowen Huang*, Shaogang Gong, Jian Zhang, Wen Gao

Abstract—Fast person re-identification (ReID) aims to search person images quickly and accurately. The main idea of recent fast ReID methods is the hashing algorithm, which learns compact binary codes and performs fast Hamming distance and counting sort. However, a very long code is needed for high accuracy (*e.g.* 2048), which compromises search speed. In this work, we introduce a new solution for fast ReID by formulating a novel Coarse-to-Fine (CtF) hashing code search strategy, which complementarily uses short and long codes, achieving both faster speed and better accuracy. It uses shorter codes to coarsely rank broad matching similarities and longer codes to refine only a few top candidates for more accurate instance ReID. Specifically, we design an All-in-One (AiO) module together with a Distance Threshold Optimization (DTO) algorithm. In AiO, we simultaneously learn and enhance multiple codes of different lengths in a single model. It learns multiple codes in a pyramid structure, and encourage shorter codes to mimic longer codes by self-distillation. DTO solves a complex threshold search problem by a simple optimization process, and the balance between accuracy and speed is easily controlled by a single parameter. It formulates the optimization target as a F_β score that can be optimised by Gaussian cumulative distribution functions. Besides, we find even short code (*e.g.* 32) still takes a long time under large-scale gallery due to the $O(n)$ time complexity. To solve the problem, we propose a gallery-size-free latent-attributes-based One-Shot-Filter (OSF) strategy, that is always $O(1)$ time complexity, to quickly filter major easy negative gallery images, Specifically, we design a Latent-Attribute-Learning (LAL) module supervised a Single-Direction-Metric (SDM) Loss. LAL is derived from principal component analysis (PCA) that keeps largest variance using shortest feature vector, meanwhile enabling batch and end-to-end learning. Every logit of a feature vector represents a meaningful attribute. SDM is carefully designed for fine-grained attribute supervision, outperforming common metrics such as Euclidean and Cosine metrics. Experimental results on 2 datasets show that CtF+OSF is not only 2% more accurate but also $5\times$ faster than contemporary hashing ReID methods. Compared with non-hashing ReID methods, CtF is $50\times$ faster with comparable accuracy. OSF further speeds CtF by $2\times$ again and upto $10\times$ in total with almost no accuracy drop.

Index Terms—Person Re-Identification, Hashing, Coarse-to-Fine, Latent Attribute, One-Shot-Filter, Computer Vision, Deep Learning

1 Introduction

Person re-identification (ReID) [1], [2], [3] aims to match images of a person across disjoint cameras, which is widely used in video surveillance, security and smart city. Many methods [2], [4], [5], [6], [7], [8], [9], [10], [11] have been proposed for person ReID. However, for higher accuracy, most of them utilize a large deep network to learn high-dimensional real-value features for computing similarities by Euclidean distance and returning a rank list by quick-sort [12]. Quick-sort of high-dimensional deep features can be slow, especially when the gallery set is large. Table 1 shows that the query time per ReID probe image increases massively with the increase of the ReID gallery size; and counting-sort [13] is much more efficient than quick-sort, in which the former has a linear complexity w.r.t the gallery size ($O(n)$) whilst the latter has a logarithm complexity ($O(nlogn)$).

Several fast ReID methods [14], [15], [16], [17], [18], [19], [20], [21] have been proposed to increase ReID speed whist retaining ReID accuracy. The common main idea is hashing, which learns a binary code instead of real-value features. To sort binary codes, the inefficient Euclidean distance and quick-sort are replaced by the Hamming-distance and counting-sort [13]. Table 2 shows that computing a Hamming distance between 2048-dimensional binary-codes is $229\times$ faster than that of a Euclidean distance between real-value features.

Different from common image retrieval tasks, which are category-level matching in a close-set, ReID is instance-level matching in an open-set (zero-shot setting). For image retrieval in the ImageNet [22], the classes of training and test sets are the same and imagery appearances of different classes diverse a lot, such as dog, car, and airplane. In contrast, the training and test ReID images have completely different ID classes without any overlap (ZSL) whilst the appearances of different persons can be very similar to subtle changes (fine-grained) on clothing, body characteristics, gender, and carried-objects. The ZSL and fine-grained characteristics of ReID require state-of-the-art hashing-based fast ReID models [21] to employ very long binary codes, *e.g.* 2048, in order to retain competitive ReID accuracy. However, the binary code length affects significantly the cost of computing Hamming distance. Table 2 shows that computing

- *Guan'an Wang (Equal Contribution) is with the School of Electronic and Computer Engineering, Peking University, China (e-mail: guan.wang0706@gmail.com).*
- *Xiaowen Huang (Equal Contribution, Corresponding Author) is with the Beijing Jiaotong University, China (e-mail: xwhuang@bjtu.edu.cn).*
- *Shaogang Gong is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, UK (e-mail: s.gong@qmul.ac.uk).*
- *Jian Zhang (Corresponding Author) and Wen Gao are with the School of Electronic and Computer Engineering, Peking University, China (e-mail: zhangjian.sz@pku.edu.cn)*
- ** means equal contribution.*

TABLE 1
ReID search time per probe image by quick-sort (real-value) and counting-sort (binary). The latter is much faster.

Gallery Size	Query Time (s) Quick-Sort	Counting-Sort
1×10^4	1.0×10^{-1}	2.7×10^{-3}
1×10^5	4.3×10^{-1}	2.7×10^{-2}
1×10^6	6.4×10^{0}	2.6×10^{-1}
1×10^7	1.1×10^{2}	2.7×10^{0}
Per Sample Complexity	- $O(nlogn)$	2.6×10^{-7} $O(n)$

TABLE 2
Comparing Euclidean- and Hamming- distances, Euclidean and longer lengths are slow to compute.

Code Length	Computation Time (s) Euclidean	Hamming
32	6.8×10^{-5}	2.4×10^{-6}
128	2.6×10^{-4}	2.8×10^{-6}
512	1.0×10^{-3}	4.4×10^{-6}
$2,048$	3.9×10^{-3}	1.7×10^{-5}

a Hamming distance between two 2048-dimensional binary codes takes 1.7×10^{-5} seconds, which is $7\times$ slower than computing that of 32-dimensional binary codes at 2.4×10^{-6} seconds. This motivates us to solve the following problem: How to yield higher accuracy from hashing-based ReID using shorter binary codes.

To that end, we propose a novel Coarse-to-Fine (CtF) search strategy for faster ReID whilst also retaining competitive accuracy. At test time, our model (CtF) first uses shorter codes to coarsely rank a gallery, then iteratively utilises longer codes to further rank selected top candidates where the top-ranked candidates are defined iteratively by a set of Hamming distance thresholds. Thus, the long codes are only used for a decreasingly fewer matches in ranking in order to reduce the overall search time whilst retaining ReID accuracy. This is an intuitively straightforward idea but not easily computable for ReID due to three difficulties: (1) Coarse-to-fine search requires multiple codes of different lengths. Asymmetrically, computing them with multiple models is both time-consuming and sub-optimal. (2) The coarse ranking must be accurate enough to minimise missing true-match candidates in fine-grained ranking whilst keeping their numbers small, thus reduce the total search time. Paradoxically, shorter codes perform much worse than longer codes in ReID task therefore hard to be sufficiently accurate. (3) The set of distance thresholds for guiding the coarse search affect both final accuracy and overall speed. How to determine *automatically* these thresholds to balance optimally accuracy and speed is both important and non-trivial.

In this work, we propose a novel All-in-One (AiO) module together with a Distance Threshold Optimization (DTO) algorithm to simultaneously solve these three problems. The AiO module can simultaneously learn and enhance multiple codes of different lengths in a single model. It progressively learns multiple codes in a pyramid structure, where the knowledge from the bottom long code is shared by the top short code. We promote shorter codes to mimic longer codes by both probability- and similarity- distillation. This makes shorter codes more powerful without importing extra teacher networks. The DTO algorithm solves a complex threshold search problem by a simple optimization process and the balance between search accuracy and speed is easily controlled by a single parameter. It explores a F_β score as the optimization target formulated as Gaussian cumulative distribution functions. So that we can estimate its parameters by the statistics of Gaussian probability distributions modeling the distances of positive and negative pairs. Finally, by

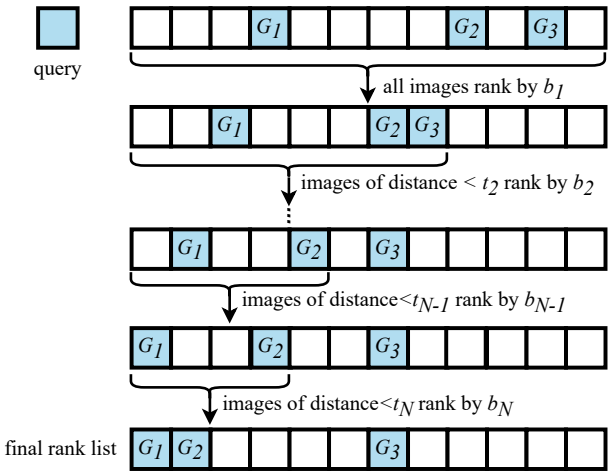

Fig. 1. Illustration of coarse-to-fine (CtF) search strategy. A Coarse-to-Fine (CtF) hashing code search strategy to speed up ReID, where Q is a query image, $\{G_i\}_{i=1}^3$ are the positive images in the gallery set, $B = \{b_k\}_{k=1}^N$ are binary codes of lengths $L = \{l_k\}_{k=1}^N$, $T = \{t_k\}_{k=2}^N$ are Hamming distance thresholds where gallery images are selected by each t_k for further comparison by increasingly longer codes b_k.

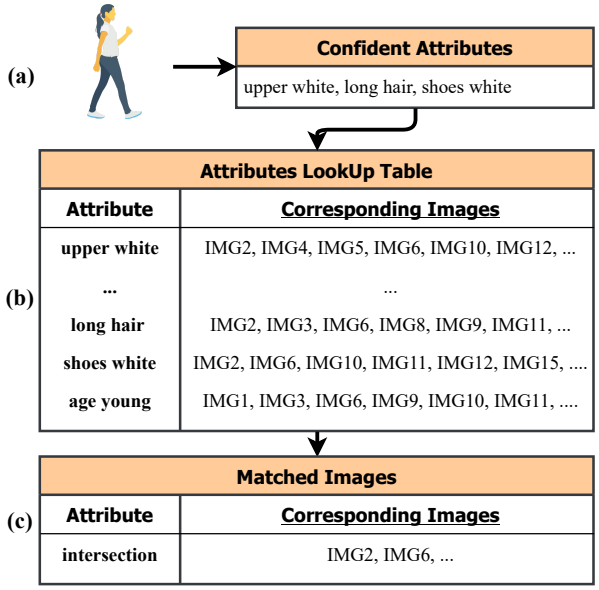

Fig. 2. Illustration of one-shot-filter (OSF) strategy. (a) Given a query image, predict its attributes and select top 3 most confident attributes. (b) Construct attributes lookup table of gallery images. This stage is done offline. (c) Index gallery images that have the 3 selected attributes. Since the attribute lookup table can be constructed offline and indexing operation is very fast under some popular database softwares, such as MySQL, one-shot-filter significantly speed Re-ID up.

TABLE 3
Brief introduction to the proposed method. The proposed method includes two core strategies: CtF and OSF. OSF filters easy negative gallery samples and CtF ranks remaining gallery ones. To kindly support the two strategies, we design AiO + DTO and LAL + SDM, respectively.

Method	Two Strategies	Key Components
Faster ReID *speeds ReID up by 10×* *Figure 5*	Coarse-to-Fine (CtF) Search: *long-short code retrieval* *speeds ReID up by 5×* *Figure 1, Algroithm 2*	All-in-One (AiO) Module: *enhance short code upto 40%* *Figure 3*
		Distance Threshold Optimization (DTO) Algrithom: *find optimal thresholds with complexity O(1)* *Algorithm 1*
	One-Shot-Filter (OSF) Strategy: *retrieval by attribute indexing* *further speeds ReID up by 2×* *Figure 2, Algorithm 3*	Latent-Attribute-Learning (LAL) Module: *batch training and end-to-end optimization for PCA* *Figure 4*
		Single-Direction-Metric (SDM) Loss: *differentiable Jaccard Metric for attributes* *Equation 18*
Test Details	Given a query, OSF firstly filters easy negative gallery samples with latent attributes, then CtF ranks remaining gallery ones with mixed long-short binary codes	

maximizing the F_β score, we compute iteratively optimal distance thresholds.

Although the proposed CtF significantly speeds retrieval up by reducing the distance computation times of longer codes, it still requires computing distances between short codes. Specifically, the time complexity of computing distance between m queries and n galleries is $O(mn)$. This also affects the retrieval speed a lot when mn is very large. The discussion above inspires us to find a way in that totally the distance computation can be avoided, thus the complexity can be dramatically reduced to $O(1)$. An intuitive idea is retrieving with semantic attributes (such as clothes color, carrying, gender). Constructing a look-up table where keys are attributes and values are corresponding images . Thus, retrieval by ranking is upgraded to retrieval by indexing[1]. However, this solution asks for accurate and generalizable semantic attributes prediction, which is not always accessible in practical. Besides, training a attribute model is also expensive. Another solution is to utilise logits in an image feature vector as attributes. For example, a 2048-dimensional feature vector may indicates 2048 attributes. However, feature vectors learned by common embedding layer (such as Linear layer) [23] and identity loss [2], [10] leads to dense knowledge (huge and fined-grained characteristics) and bidirectionally-activation features Attributes prefers to sparse knowledge (a few but significant characteristics) and single-directional attributes (True or False).

To overcome the challenges above, we propose a novel Latent-Attribute-Learning (LAL) module together with a Single-Direction-Metric (SDM) Loss. LAL is derived from principal component analysis (PCA) that keeps largest variance (significant characteristics) using a shortest feature vector (a few of characteristics), meanwhile enabling batch and end-to-end learning. Every logit of a feature vector represents a meaningful attribute. SDM is carefully designed for fine-grained attribute supervision, outperforming common metrics such as Euclidean and Cosine metrics. It is based on Jaccard metric and powered by gradient computation.

1. indexing can be extremely fast under many database softwares such as Oracle, MySQL.

Our contributions can be summarised below, a brief version is displayed in Table 3.

(1) We propose a novel ReID method that speeds retrieval up whilst keeps accuracy. It consists of two main strategies, Coarse-to-Fine (CtF) and One-Shot-Filter (OSF). CtF utilises mixed long-short code search, and OSF upgrades retrieval-by-ranking with retrieval-by-indexing. Given a query, OSF first filters very easy negative gallery samples, and then CtF ranks the remaining gallery samples.

(2) The Coarse-to-Fine (CtF) strategy includes an All-in-One (AiO) module and a Distance Threshold Optimization (DTO) algorithm. The AiO module learns multiple codes of different lengths in a pyramid structure and enhances them via probability- and similarity-distillation loss. The DTO algorithm finds the optimal thresholds for coarse-to-fine search by concluding the threshold search task to a F_β distance optimization problem.

(3) The One-Shot-Filter (OSF) strategy consists of a Latent-Attribute-Learning (LAL) module and a Single-Direction-Metric (SDM) loss. The LAL module automatically learns potential attributes with only identity labels not attribute labels. It is derived from principal component analysis (PCA) and enables batch and end-to-end learning. The SDM loss is an IOU-like metric, which is derived from the Jaccard metric and powered by gradient computation, outperforms common Euclidean and Cosine metrics.

(4) Extensive experimental results on Market-1501 and DukeMTMC-ReID datasets show that our proposed CtF is $50\times$ faster than non-hashing ReID methods, $5\times$ faster and 2% more accurate than hashing ReID methods. OSF further speeds CtF up by $2\times$ and upto $10\times$ in total with almost no accuracy drop. Experiments on MSMT also validate its effectiveness on large-scale dataset. Besides, experiments on different baselines show its scalability to different backbones.

2 Related Works

In this paper, we try to solve the fast ReID task under the framework of hashing by proposing an All-in-One (AiO)

hashing learning module and a Distance Threshold Optimization (DTO) algorithm. Thus, we mainly discuss the related works including non-fast person re-identification (ReID) task, fast ReID task and hashing algorithm.

2.1 Person Re-Identification

Person re-identification addresses the problem of matching pedestrian images across disjoint cameras [1]. The key challenges lie in the large intra-class and small inter-class variation caused by different views, poses, illuminations, and occlusions. Existing methods can be grouped into hand-crafted descriptors [4], [5], [6], metric learning methods [7], [8], [9] and deep learning algorithms [2], [10], [11], [24], [25], [26], [27], [28], [29]. The goal of hand-crafted descriptors is to design robust features. Metric learning aims to make a pair of true matches have a relatively smaller distance than that of a wrong match pair in a discriminant manner. Deep learning algorithms adopt deep neural networks to straightly learn robust and discriminative features in an end-to-end manner and achieve the best performance. Here, we mainly show some deep learning methods. For example, Zheng *et al.* [2] learn identity-discriminative features by fine-tuning a pre-trained CNN to minimize a classification loss. In [10], Hermans *et al.* show that using a variant of the triplet loss outperforms most other published methods by a large margin. In [11], a network named Part-based Convolutional Baseline (PCB) is proposed to learn fine-grained part-level features with a uniform partition strategy. However, all the ReID methods above learn real-value features for high accuracy, which is slow.

2.2 Hashing Algorithm

Hashing algorithm mainly divided into unsupervised and (semi-)supervised ones. Unsupervised hashing methods (LSH [30], SH [31], ITQ [32]) employ unlabeled data even no data. (Semi-)Supervised (KSH [33], SDH [34]) utilize labeled information to improve binary codes. Recently, inspired by powerful deep networks, some deep hashing methods (CNNH [35], DPSH [36], SSGAH [37], ABML [38],) have been proposed and achieve much better performance. They usually utilize a CNN to extract meaningful features, formulate the hashing function as a fully-connected layer with $tanh/sigmoid$ activation function, and quantize features by $signature$ function. The framework can be optimized with a related layer or some iteration strategies. However, all the hashing methods are designed for close-set category-level retrieval tasks, which cannot be directly used for person ReID, an open-set fine-grained search problem.

2.3 Fast Person Re-Identification

Fast ReID methods aims to search in a fast speed meanwhile obtaining accuracy as high as possible. The main idea of those methods is hashing algorithm, which learns binary code instead of real-value features. Based on the binary codes, the inefficient Eulidean distance and quick-sorting can be replaced by efficient Hamming distance and counting sort. Zheng *et al.* [15] learn cross-view binary codes using two hash functions for two different views. Wu *et al.* [16] simultaneously learn both CNN feature and hash functions to get robust yet discriminative features and similarity-preserving binary codes. CSBT [18] solves the cross-camera variations problem by employing a subspace projection to maximize intra-person similarity and inter-person discrepancies. In [17] integrate spatial information for discriminative features by representing horizontal parts to binary codes. ABC [21] improves binary codes by implicitly fits the feature distribution to a pre-defined binary one with Wasserstein distance. However, all the fast ReID methods take very long binary codes (*e.g.* 2048) for high accuracy. Different from them, we propose a coarse-to-fine search strategy which complementarily uses codes of different lengths, obtaining not only faster speed but also higher accuracy.

3 Proposed Method

In this paper, we propose a novel fast Re-ID method for fast and accurate ReID, which includes two core ideas, *i.e.* one-shot-filter (OSF) and coarse-to-fine (CtF) search strategies. The former filters major easy negative gallery samples using attributes. To flexibly and accurately learn attributes, a Latent-Attribute-Learning (LAL) module together with a Single-Direction-Metric (SDM) loss are proposed to learn without manual annotation. The latter efficiently search the remaining gallery samples with mixed shot-long binary codes. We design an All-in-One (AiO) module together with a Distance Threshold Optimization (DTO) algorithm. The AiO learns and enhances multiple codes of different lengths in a single module. The latter finds the optimal distance thresholds to balance time and accuracy with time complexity $O(1)$. The CtF speeds ReID up by $5\times$ and the OSF further speeds the CtF up by $2\times$, getting a $10\times$ faster speed in final.

3.1 Coarse-to-Fine Search

As we illustrated in the introduction section, although the long binary codes can get high accuracy, it takes much longer time than short codes. This motivates us to think about that can we reduce the usage of long codes to further speed hashing ReID methods up. Thus, a simple but efficient solution is complementarily using both short and long codes. Here, shorter codes fast return a rough rank list of gallery, and longer codes carefully refine a small number of top candidates. Figure 1 show its procedures. Although the idea is straightforward, as discussed in paragraph 4 of section 1, there are three difficulties blocking the idea. To solve the problems, we propose an All-in-One (AiO) module and a Distance Threshold Optimization (DTO) algorithm. Please see the next two parts for more details.

3.1.1 All-in-One Module

The All-in-One (AiO) module aims to simultaneously learn and enhance multiple codes of different lengths in a single model, whose architecture can be seen in Figure 3. Specifically, it first utilizes a convolutional network to extract the real-value feature vectors, then learns multiple codes of different lengths in a pyramid structure, finally enhances the codes by encouraging shorter codes mimic longer codes via self-distillation.

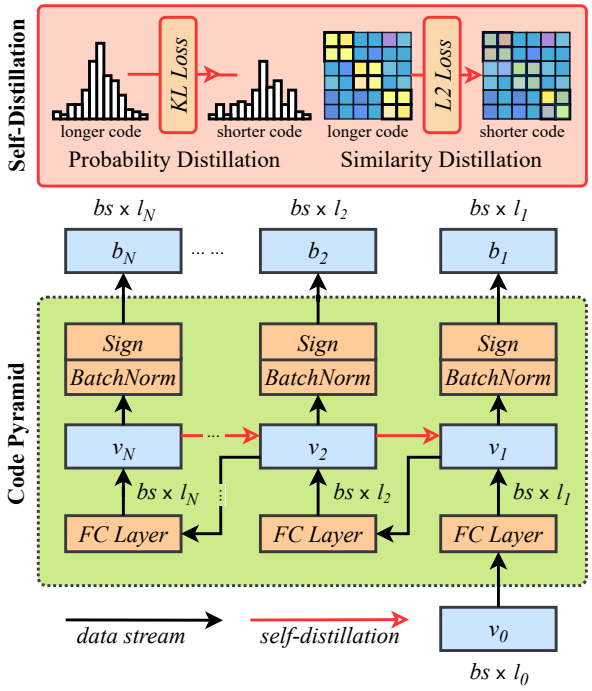

Fig. 3. All-in-One (AiO) module. It learns and enhances multiple codes of different lengths in a single module with a code pyramid structure and self-distillation learning.

Learn Multiple Codes in a Pyramid Structure. The code pyramid learns multiple codes of different lengths, where the shorter codes are based on the longer codes. With such a structure, we can not only learn many codes in one shot, but also share the knowledge of longer codes with shorter codes. The equations are as below:

$$v_0 = F(x), \;\; v_k = FC_k(v_{k-1}), \;\; k \in 1, 2, ..., N, \qquad (1)$$

where x is input image, F is the CNN backbone, N is the code number, $V = \{v_k\}_{k=1}^N$ are the real-value feature vectors with different lengths $L = \{l_k\}_{k=1}^N$, FC_k is the fully-connected layers with l_{k-1} input- and l_k output-sizes. After getting real-value features of different lengths, we can obtain their binary codes $B = \{b_k\}_{k=1}^N$ in the following equation.

$$b_k = sign(bn(v_k)), \qquad (2)$$

where bn is the batch normalization layer, $sign$ is the symbolic function. We use the batch normalization layer because it normalizes the real-value features to be symmetric to 0 and reduces the quantization loss.

Enhance Codes with Self-Distillation Learning. As we discussed in the introduction section, the coarse ranking must be accurate enough to minimise missing true-match candidates in fine-grained ranking. Inspired by [39], [40], we introduce self-distillation learning to enhance the multiple codes in a single module without importing extra teacher network. Different from conventional distillation models, which imports an extra large teacher network to supervise a small student network, we perform distillation learning in a single network and achieve better performance, which is important for fast ReID.

Specifically, our self-distillation learning is composed of a probability- and a similarity- distillation. The probability-distillation transfers the instance-level knowledge in a from of softened class scores. Its formulation is given by

$$\mathcal{L}_{pro} = \frac{1}{N-1} \sum_{k=1}^{N-1} \mathcal{L}_{ce}(\sigma(\frac{z_{k+1}}{T}), \sigma(\frac{\hat{z}_k}{T})), \qquad (3)$$

where $\mathcal{L}_{ce}(\cdot,\cdot)$ denotes the cross-entropy loss, σ is the softmax function, $\hat{z}_k/z_{k+1}$ means the output logits of the binary code b_k/b_{k+1}, $\hat{z}_k$ means it act as a teacher and fixed during training, T is a temperature hyperparameter, which is set 1.0 empirically. The similarity-distillation transfers the knowledge of relationship from longer codes to shorter one, whose formulation is in Eq.(4). This is motivated by that as an image search task, ReID features should also focus on the relationship among samples, *i.e.* to what extent the sample A is similar/dissimilar to sample B.

$$\mathcal{L}_{sim} = \frac{1}{N-1} \sum_{k=1}^{N-1} \sum_{i,j} || \frac{1}{l_{k+1}} G_{k+1}^{i,j} - \frac{1}{l_k} \hat{G}_k^{i,j} ||^2, \qquad (4)$$

where $G_k^{i,j}/G_{k+1}^{i,j}$ is the Hamming distance between b_k^i/b_{k+1}^i and b_k^j/b_{k+1}^j, $b_{k/k+1}^{i/j}$ is the binary code of image x_i/x_j with length l_k/l_{k+1}, the $\hat{G}$ means that G acts as a label and is fixed during the optimization process, thus contributes nothing to the gradients.

Overall Objective Function and Training. Recent progresses on ReID have shown the effectiveness of the classification [2] and triplet [10] losses. Thus, our final objective function includes our proposed probability- and similarity-distillation losses together with the classification and triplet losses as the final objective function. The formulation can be found in Eq.(5),

$$\mathcal{L}_{ctf} = \mathcal{L}_{ce} + \mathcal{L}_{tri} + \lambda_{prob}\mathcal{L}_{prob} + \lambda_{sim}\mathcal{L}_{sim} \qquad (5)$$

Considering that the mapping function sgn in Eq.(2) is discrete and Hamming distance in Eq.(2) is not differentiable, a natural relaxation [36] is utilised in Eq.(5) by replacing sgn with $tanh$ and changing the Hamming distance to the inner-product distance. Finally, our All-in-One module can be optimized in an end-to-end way by minimizing the loss in Eq.(5).

Algorithm 1. Distance Threshold Optimization

Input: Trained Model in Eq.(2), Validation Data (X_v, Y_v)
Output: Thresholds $\{T_i\}_{i=2}^N$

for $k = \{1, 2, ..., n-1\}$ **do** [a]
  B_k: Extract binary codes with length l_k via Eq.(2)
  D^r: Hamming distances of positive pairs (b_k^i, b_k^j)
  D^n: Hamming distances of negative pairs (b_{k-1}^i, b_{k-1}^j)
  PDF^r, PDF^n: PDF of D^r and D^n of in Eq.(7)
  CDF^r, CDF^n: CDF of D^r and D^n in Eq.(7)
  t_{n+1}: Maximize F_β score in Eq.(8) and return t_{n+1}
return $T = \{t_i\}_{i=2}^N$

a. $y^i = y^j$ in positive pairs, $y^i \neq y^j$ in negative pairs, PDF is probability distribution function, CDF is cumulative Distribution Function.

3.1.2 Distance Threshold Optimization

After getting the multiple codes of different lengths $B = \{b_i\}_{i=1}^N$, we can perform the Coarse-to-Fine (CtF) search.

There are two tips in CtF search, *i.e.* high accuracy and fast speed. For fast speed, the candidate number returned by coarse search should be small. For high accuracy, the candidates returned by coarse search should include relevant images as more as possible. But the two requirements are naturally conflicting. Thus, it is important to find the proper thresholds to optimally balance the two targets, *i.e.* both high accuracy and fast speed. One simple solution is brute search via cross-validation. However, the search space is too large. For example, if we have multiple binary codes of lengths $L = \{32, 128, 512, 2048\}$, the complexity of the brute search will be $\prod_L > 4 \times 10^9$ times.

In this part, we propose a novel Distance Threshold Optimization (DTO) algorithm which solves the time-consuming brute parameter search task with a simple optimization process. Specifically, inspired by [41], we first explicitly formulate the two sub-targets as two scores in Eq.(6), *i.e.* precision (P) and recall (R) scores. Then we balance the two sub-targets by mixing the two scores with a single parameter β and get F_β score in Eq.(6).

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F_\beta = (\beta^2+1)\frac{PR}{\beta^2 P + R} \tag{6}$$

Here, TP is the number of relevant images in the candidates, FP is the number of non-relevant images in the candidates and FN is not retrieved relevant samples. As we can see, the precision score P means the rate of relevant images in the candidates. Usually a high P means a small candidate number, which is good for fast speed. The recall score R represents the rate of returned relevant samples in the total relevant samples. A high R score means more returned relevant samples, which is important for high accuracy. The F_β mixed the precision and recall scores with a parameter β, which considers both speed and accuracy.

$$\begin{aligned} PDF(t) &= \frac{1}{\sigma\sqrt{2\pi}} exp(-\frac{(t-u)^2}{\sigma\sqrt{2}}) \\ CDF(t) &= \frac{1}{2}(1 + erf\frac{t-u}{\sigma\sqrt{2}}) \end{aligned} \tag{7}$$

$$F_\beta = \frac{CDF^r(\beta^2+1)}{CDF^n + CDF^r + \beta^2(1 - CDF^n + CDF^r)} \tag{8}$$

Considering that TP/FP/FN are statistics which cannot be optimized, we replace them with two Gaussian cumulative distribution functions in form of Eq.(7) (right), whose parameters u and σ are estimated by fitting a validation set using the Gaussian probability distribution function in Eq.(7) (left). Finally, by maximizing the F_β in Eq.(8), we can get the optimal distance thresholds $T = \{t_k\}_{k=2}^N$ balanced by β.

3.1.3 Summary of CtF

In the training stage, we minimize $\mathcal{L}_{ctf}$ in Eq.(5). In the testing stage, the details are summarised in Algorithm 2.

Algorithm 2. Coarse-to-Fine Strategy

Input: a Query Data x_q, a set of Gallery Data $X_g = \{x_i\}_{i=1}^{N_g}$, Trained AiO Module in Eq.(2), Thresholds $\{T_i\}_{i=2}^{N}$
Output: Ranked Gallery Data $\hat{X}_g = \{\hat{x}_i\}_{i=1}^{N_g}$

X_{kpt}: Initialize kept gallery data as X_g
$\hat{X}_g$: Initialize ranked gallery data as X_{kpt}
for $k = \{1, 2, ..., n-1\}$ **do**
    D_k: Hamming distances between x_q and X_{kpt} under code length l_k
    X_{kpt}: Rank X_{kpt} with D_k in ascend
    $\hat{X}_g$: Record rank results with $\hat{X}_g[: len(X_{kpt})] = X_{kpt}$
    X_{kpt}: Select gallery data to be refined with $X_{kpt} = X_{kpt}[: np.argwhere(X_{kpt} < T_k)]$
return $\hat{X}_g$

3.2 One-Shot Filter

As we illustrated in the introduction section, although the Coarse-to-Fine (CtF) search strategy significantly speeds retrieval up meanwhile keeps high accuracy, it still gets an $O(mn)$ time complexity for distance computation where m and n are query and gallery sizes, respectively. This motivates us to find a way in which total distance computation is avoided and an $O(1)$ time complexity is obtained.

One intuitive idea is upgrading retrieval-by-ranking problem to a retrieval-by-indexing problem, where a look-up table can be constructed and some advanced databases (Oracle, MySQL) are naturally utilised to speed retrieval up. For example, given an male, all female can be filtered. However, this idea requires accurate and generalizable attribute prediction. Consequently, lots of attribute annotation is needed, which limits its flexibility. An alternative is viewing every logit of an identity-feature vector as a latent attribute. For example, a 2048-dimensional feature may indicates 2048 attributes. However, existing Re-ID models learn identity-features in metric of Euclidean and Cosine, which is not suitable for attribute representation. The former represents fine-grained information (such as texture) with dense features (such as 2048-dimension). The latter only asks for a few of remarkable information (such as gender) with sparse attributes (e.g. 27 attributes for Market-1501).

To solve the problem above, we propose a Latent-Attribute Learning (LAL) module and a Single-Direction-Metric (SDM) loss. The former formats latent-attribute learning problem as a feature decomposition procedure in sphere space, naturally getting sparse, principal and explainable latent attributes. The latter optimizes latent-attribute in a Jaccard metric, outperforming either Euclidean or Cosine metrics. Please see corresponding section for details.

3.2.1 Latent-Attribute Learning Module

The Latent-Attribute Learning (LAL) module is inspired by the principal component analysis (PCA). We first review PCA and then adapt it to our task.

Review Principal Component Analysis (PCA). The PCA is defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. The formulation of PCA is as below:

$$\begin{aligned} X^{out} &= X^{in}U^T \\ s.t.\ \ \Lambda &= U\Sigma U^T \\ I &= UU^T \end{aligned} \tag{9}$$

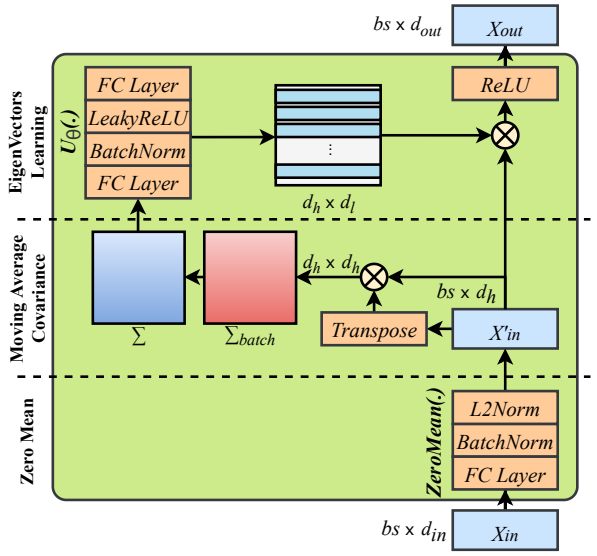

Fig. 4. Latent-Attribute Learning (LAL) module. It learns latent attributes with the largest variance. X_{in} and X_{out} mean the input feature and output attributes, respectively. $X'_{in} = ZeroMean(X_{in})$ is input feature with zero mean.

In the equation above, $X^{in} \in R^{n \times d_l}$ is a set of zero-meaned features with long dimension, $X^{out} \in R^{n \times d_s}$ is a set of features with short dimension, $\Sigma \in R^{d_l \times d_l}$ is the covariance of long features, $U \in R^{d_s \times d_l}$ is the top d_s eigenvectors, $\Lambda \in R^{d_s \times d_s}$ is the diagonal matrix with eigenvalues in diagonal elements and zeros in non-diagonal elements.

Adapt to Batch Learning with Moving Average Covariance. Inspired by PCA, which reduces several long features into several short features whilst keeping the greatest variance, a short feature vector can be viewed as several latent attribute logits and U can be viewed as latent attributes d_s. However, U is post-calculated from a global statistic Σ, which fails to deal with end-to-end optimization and batch training. In this part, we propose a novel Latent-Attribute Learning (LAL) modules to learn a latent attribute, which keep the least attributes, end-to-end optimizable, batch trainable. The LAL module contributes to two key improvements, (1) enabling end-to-end optimization by replacing U with a trainable function $U_\theta(\Sigma)$, (2) enabling batch training with moving average covariance $\Sigma = \eta\Sigma + (1-\eta)\Sigma_{batch}$, where η is set 0.9 empirically.

Adapt to End-to-End Learning with Trainable Eigenvectors. The detailed structure of LAL module is displayed in Figure 4. It includes a zero mean part, moving average covariance part, an eigenvector learning part. The zero-mean part $ZeroMean$ consists of a linear layer from 2048 to 512, a batch normalization layer with untrainable parameters (weight and bias are set 1 and 0, respectively), and a normalization layer which normalizes every feature to be norm 1. The moving average covariance part first compute covariance Σ_{batch} within a batch, then calculate moving average covariance Σ with strategy $\Sigma = \eta\Sigma + (1-\eta)\Sigma_{batch}$. The eigen-vectors learning part utilise a function $U_\theta(\cdot)$, which consists of a linear layer, a batch normalization layer, a leakyReLU layer with ratio 0.1 and a linear layer. It predicts eigenvectors given moving average covariance, *i.e.* $U = U_\theta(\Sigma)$. Besides, a ReLU layer is used to constrain value to be alway positive. The final formulation of LAL module is as below:

$$\begin{aligned} X^{out} &= ReLU(ZeroMean(X^{in})U_\theta(\Sigma)^T) \\ s.t.\ \ \Sigma &= U_\theta(\Sigma)^T \Lambda_\Phi U_\theta(\Sigma) \\ I &= U_\theta(\Sigma)U_\theta(\Sigma)^T \end{aligned} \tag{10}$$

where Λ_Φ is a square matrix with trainable diagonal elements and the others zero.

Objective Function of LAL module. The final LAL module converts the two constraints to be two losses including an identity loss $\mathcal{L}_{identi}$ and an orthogonality loss $\mathcal{L}_{orth}$.

$$\begin{aligned} L_{identi} &= ||I - U_\theta(\Sigma)U_\theta(\Sigma)^T||_2 \\ \mathcal{L}_{orth} &= ||\Sigma - U_\theta(\Sigma)^T \Lambda_\Phi U_\theta(\Sigma)||_2 \end{aligned} \tag{11}$$

Discussion. The proposed LAL module is derived from principal component analysis (PCA) and powered by batch learning and trainable eigenvector abilities. The batching learning ability is similar to incremental PCA [42], which estimates the top eigenvector incrementally. However, the incremental PCA still computes the eigenvector in a statistical way, which is indifferentiable and fails to apply to a deep learning pipeline. Our proposed LAL module is specifically designed for latent attribute learning in a deep-learning pipeline and a novel single-direction-metric loss is proposed, which guarantee its optimization with stochastic gradient descent.

3.2.2 Single-Direction-Metric Loss

The LAL module would like to learn latent attributes. However, existing metrics (*e.g.* Euclidean, Cosine) fail to deal with attributes, that require an IOU-like metric. Here, we utilise Jaccard Similarity to metric attributes and improve it to be an end-to-end version. This part is inspired by [43], [44] and improved to adapt person re-identification triplet loss.

Review Jaccard Similarity. The Jaccard Similarity measures two sets. It is defined as the size of the intersection divided by the size of the union of two sets. Given two sets $\mathbf{A}$ and $\mathbf{B}$, the Jaccard Similarity is computed using the following formula:

$$\mathcal{J}(\mathbf{A}, \mathbf{B}) = \frac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A} \cup \mathbf{B}|}, \tag{12}$$

where $|\cdot|$ denotes the cardinality of a set. Using C-dimension binary vectors $\{0,1\}^C$ to represent set $\mathbf{A}$ and $\mathbf{B}$, where each channel denotes a specific attribute, 1 means activated attribute and 0 means deactivated attribute, the Jaccard Similarity between these two sets is computed by:

$$\mathcal{J}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{c=1}^{C} \mathbf{A}[c] \wedge \mathbf{B}[c]}{\sum_{c=1}^{C} \mathbf{A}[c] \vee \mathbf{B}[c]}, \tag{13}$$

where $\wedge$ and $\vee$ denote logic-AND and logic-OR opearators, and the operators $[\cdot]$ return the element at the position c in the attribute set. To make the Jaccard Similarity adaptive to this continuous variable, we use minimization and maximization to approximate the bit-wise AND and OR operators in Eq.(14), respectively. For given two attribute sets g_1 and g_2, the Jaccard Similarity is redefined by:

$$\mathcal{J}(\mathbf{g}_1, \mathbf{g}_2) = \frac{\sum_{c=1}^{C} min(\mathbf{g}_1[c], \mathbf{g}_2[c])}{\sum_{c=1}^{C} max(\mathbf{g}_1[c], \mathbf{g}_2[c])}, \tag{14}$$

Algorithm 3. *O*ne-Shot Filter

Input: Trained LAL module $U_\theta(\cdot)$ in Eq.(10), a Query Data x_q, a set of Gallery Data $X_g = \{x_i\}_{i=1}^{N_g}$, Filter Threshold γ

Output: Kept Gallery Data $\hat{X}_g = \{x_i\}_{i=1}^{\hat{N}_g}$, $\hat{N}_g$ is the number of kept data, $\hat{N}_g < N_g$

Construct Latent Attribute LookUpTable (Offline)

$\mathbf{G}_g = \{\mathbf{g}_i\}_{i=1}^{N_g}$: Extract latent attribute vectors of gallery data X_g via LAL, $\mathbf{G}_g \in [0,1]^{N_g \times C}$, C is attribute number
LookUpTable: Init LookUpTable as a dictionary
for $c = \{1, 2, ..., C\}$ **do**
    LookUpTable[c] = np.argwhere($\mathbf{G}_g[:, c] > 0$).tolist()

Filter with Latent Attribute LookUpTable (Online)

$\mathbf{g}_q$: Extract the latent attribute vector of the query data x_q via LAL, $\mathbf{g}_q \in [0,1]^C$
$\mathbf{a}_q$ = np.argsort($\mathbf{g}_q$, order='descend')[:γ].tolist(): get top γ activated attribute indexes of the query data x_q
$\hat{X}_g$ = intersection([LookUpTable[i] for i in $\mathbf{a}_q$]): find gallery data whose $\mathbf{a}_q$ attributes are all activated
Return: $\hat{X}_g$

where c denotes the attribute index. Further, to smooth the min/max operators, we introduce a Softmax-Jaccard Similarity:

$$\mathcal{J}(\mathbf{g}_1, \mathbf{g}_2) = \frac{\sum_{c=1}^{C}(\mathbf{w}_1^{min}[c] \cdot \mathbf{g}_1[c] + \mathbf{w}_2^{min}[c] \cdot \mathbf{g}_2[c])}{\sum_{c=1}^{C}(\mathbf{w}_1^{max}[c] \cdot \mathbf{g}_1[c] + \mathbf{w}_2^{max}[c] \cdot \mathbf{g}_2[c])}, \tag{15}$$

$$\mathbf{w}_k^{min}[c] = \frac{e^{-\tau \cdot \mathbf{g}_k[c]}}{\sum_{n=1}^{N} e^{-\tau \cdot \mathbf{g}_n[c]}}, \mathbf{w}_k^{max}[c] = \frac{e^{\tau \cdot \mathbf{g}_k[c]}}{\sum_{n=1}^{N} e^{\tau \cdot \mathbf{g}_n[c]}}, \tag{16}$$

where, $\mathbf{w}_k^{min}/\mathbf{w}_k^{max}$ is softmin/softmax of $\mathbf{g}_k[c]$ along k: $k = 1, 2, ..., N$, N is the batch size, τ is the smoothing factor.

Normalization. To keep the $J_s(\cdot,\cdot)$ belong to range $[0, 1]$, we normalize $\mathbf{w}_k^{min}/\mathbf{w}_k^{max}$ when computing $\mathbf{g}_i$ and $\mathbf{g}_j$:

$$\begin{aligned} \mathbf{w}_i^{min}[c] = \frac{\mathbf{w}_i^{min}[c]}{R^{min}}, \mathbf{w}_j^{min}[c] = \frac{\mathbf{w}_j^{min}[c]}{R^{min}}, \\ s.t.\ \ R^{min} = \mathbf{w}_i^{min}[c] + \mathbf{w}_j^{min}[c], \\ \mathbf{w}_i^{max}[c] = \frac{\mathbf{w}_i^{max}[c]}{R}, \mathbf{w}_j^{max}[c] = \frac{\mathbf{w}_j^{max}[c]}{R^{max}}, \\ s.t.\ \ R^{max} = \mathbf{w}_i^{max}[c] + \mathbf{w}_j^{max}[c]. \end{aligned} \tag{17}$$

Single-Direction-Metric (SDM) loss. SDM loss is defined in the equation below, where g_n means the attribute of sample x_n, g_{n-} is attribute of a negative sample of x_n, g_{n+} belongs to a positive sample of x_n, δ is a margin parameter.

$$\mathcal{L}_{sdm} = \sum_{n=1}^{N}[\delta + \mathcal{J}(g_n, g_{n+}) - \mathcal{J}(g_n, g_{n-})]_+, \tag{18}$$

3.2.3 Summary of OSF

This section summarises the training and testing details in Eq.(19) and Algorithm 3, respectively.

Training Stage. In the training stage, overall objective function of the proposed One-Shot-Filter (OSF) strategy is shown in the equation below, where λ_* are corresponding weights.

$$\mathcal{L}_{osf} = \lambda_{sdm}\mathcal{L}_{sdm} + \lambda_{identi}\mathcal{L}_{identi} + \lambda_{orth}\mathcal{L}_{orth} \tag{19}$$

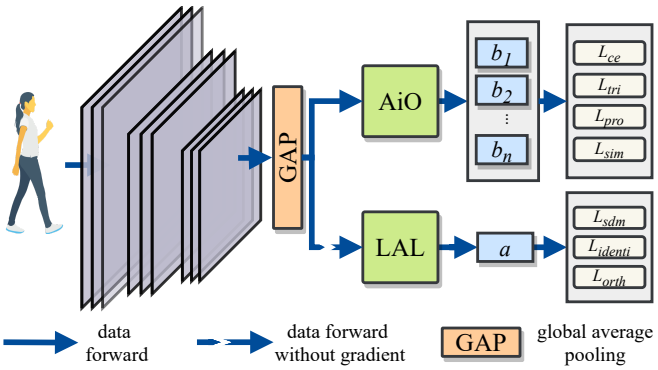

Fig. 5. Overview of our proposed method. AiO and LAL are All-in-One (Figure 3) and Latent-Attributes-Learning (Figure 4) modules, respectively. L_* means losses in Eq.(5) and Eq.(19), respectively.

Testing Stage. In the testing stage, as shown in Algorithm 3, OSF includes two steps: (1) offline construct latent attribute look-up table and (2) online filter negative samples with the latent attribute look-up table. Specifically, given a trained latent-attribute-learning (LAL) module in Eq.(10), a query data x_q, a set of gallery data $X_g = \{x_i\}_i^{N_g}$ and filter threshold γ, OSF first constructs a look-up table, whose keys are attribute indexes and values are corresponding gallery data indexes. The look-up table is only initialized one time and reused for all queries. Then, given a query data, OSF extracts its attribute vector, selects top γ most confident ones, and finds gallery data that own all the γ attributes activated. γ is a hyper-parameter to balance accuracy and speed. A larger γ filters more negative images, contributing to faster speed, but may discard more positive samples, harming accuracy. Oppositely, a smaller γ guarantees accuracy with less speed improvement. We set $\gamma = 1$ via cross-validation.

3.3 Overall Framework

Overall framework of our proposed method is shown in Figure 5. A convolutional neural networks (CNN) module together with a global average pooling (GAP) extracts feature vectors of input images. Following that, two branches, *i.e.* AiO and LAL together with corresponding losses learn binary codes and latent attributes. Its objective function is shown below:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{tri} + \lambda_{prob}\mathcal{L}_{prob} + \lambda_{sim}\mathcal{L}_{sim} \\ + \lambda_{sdm}\mathcal{L}_{sdm} + \lambda_{identi}\mathcal{L}_{identi} + \lambda_{orth}\mathcal{L}_{orth} \end{aligned} \tag{20}$$

In testing stage, given a query image and a set of gallery images, OSF is first utilised to filter major simple negative samples, then CtF gradually ranks remaining gallery samples with mixed long-short code.

4 Experiments

4.1 Dataset and Evaluation Protocols

Datasets. We extensively evaluate our proposed method on two common datasets (Market-1501 [45] and DukeMTMC-reID [46]) and one large-scale dataset (Market-1501+500k [45]). The Market-1501 dataset contains 1,501 identities observed under 6 cameras, which are splited into 12,936 training, 3,368 query and 15,913 gallery images. The Market-1501+500k enlarges the gallery of Market-1501 with extra

500,000 distractors, making it more challenging for both accuracy and speed. DukeMTMC-reID contains 1,404 identities with 16,5522 training, 2,228 query and 17,661 gallery images.

Evaluation Protocols. For accuracy, we use standard metrics including Cumulative Matching Characteristic (CMC) curves and mean average precision (mAP). All the results are from a single query setting. To evaluate speed, we use average query time per image, including distance computation and sorting time. For fair evaluation of query time, we do not use any parallel algorithm for distance computation and sorting.

4.2 Implementation Details

We implemented our method with Pytorch on a PC with 2.6Ghz Intel Core i5 CPUs, 10GB memory, and a NVIDIA RTX 2080Ti GPU. For a fair comparision and following most ReID methods [21], [23], we use Resnet-50 [47] as the CNN backbone. In training stage, each image is resized to 256×128 and augmented by horizontal flip and random erasing [48]. A batch data includes 64 images from 16 different persons, where every person includes 4 images. The lengths $L = \{l_k\}_{k=1}^{N}$ of multiple codes are empirically set $\{32, 128, 512, 2048\}$. The margin in the triplet loss in Eq.(5) is 0.3. The framework is optimized by Adam [49] with total epochs 120. Its initial learning rate is 0.00035, which is warmed up for 10 epochs and decayed to its $0.1\times$ and $0.01\times$ at 40 and 70 epochs. We randomly split the training data into a training and a validation set according to $6:4$, then decide the parameters via cross-validation, After that, we train our method with the whole training data. λ_{prob} and λ_{sim} in Eq.(5) are set as 1.0 and 1,000, and β in Eq.(8) is set 2.0. λ_{sdm}, λ_{identi} and λ_{orth} are set 1.0.

4.3 Comparisons with Non-Hashing ReID Methods

Non-hashing ReID use longer real-value features, such as 2048-dimensional $float64$ features, for a better accuracy. This significantly affects their speed, *i.e.* query time. Table 4 shows that our proposed CtF (including AiO) method is significantly faster than non-hashing ReID methods (two orders of magnitude). CtF also achieves very competitive accuracy with close Rank-1 (93.7% vs. 94.1%) and mAP (87.6% vs. 86.4%) scores of the very popular baseline ReID mehtod BoT [23] on Market-1501 and DukeMTMC-reID, and better than most the other non-hashing methods using different feature length, of which methods have features shorter than 2,062 (*e.g.* PSE [50], IDE [2], PN-GAN [51], CamStyle [53], PIE [73]) and methods have features longer than 10,240 (*e.g.* SPReID [61], PCB [11], VPM [63]). Overall, longer feature usually contributes to higher accuracy but with slower speed. For example, SPReID, PCB and VPM take features longer than 10,240 and achieves 92%-93% and 83%-84% Rank-1 scores on Market-1501 and DukeMTMC-reID datasets, respectively. The others utilize features no longer than 2,048 achieving Rank-1 score less than 92% and 80%. On the other hand, the query speed of those methods with long features is much slower. For example, PCB takes $6.9s$ and $6.3s$ for query each image on the two datasets respectively. This is 3-$4\times$ slower than IDE with $2s$ on either dataset. Specifically, CtF+OSF performs much faster than non-hashing methods and significantly, it achieves comparabale accuracy with real-value feature models. For example, CtF+OSF achieves $95.5\%/91.4\%$ Rank-1 scores on Market-1501/DukeMTMC-reID, as compared to LUPersonNL having $96.6\%/92.0\%$ respectively. This is because CtF (including AiO) utilizes all-in-one module together with coarse-to-fine search strategy, which not only learns powerful binary code, but also complementarily uses short and long codes for both high accuracy and fast speed. Meanwhile, OSF filters simple hard negative samples with very fast look-up table, which significantly reduces gallery size.

4.4 Comparisons with Hashing ReID Methods

Hashing ReID methods learn binary codes using a hashing algorithm. Binary codes are good for speed but sacrifice model accuracy. To mitigate this problem, the state-of-the-art hashing ReID methods usually employ long codes such as 2048. In binary coding, 2048 is relatively very long as compared to the more commonly used 512 length, unlike in real-value feature length compared above. Table 5 shows that CtF (with AiO) not only achieves the best accuracy (even compared to much shorter code length used by other hashing methods), but also is significantly faster than existing hashing ReID methods (even compared to the same code length used by other hashing methods). Overall, hashing ReID methods usually perform much worse than non-hashing methods. For example, best non-hashing ReID methods achieves 93.3% and 84.3% mAP scores on Market-1501 and DukeMTMC-reID respectively. But the best hashing ReID method only obtains 88.8% and 79.4% Rank-1 scores. Moreover, existing hashing ReID models can increase accuracy by using longer code length and compromising speed. For example, ABC with 512-dimensional binary codes achieves $69.4\%/69.9\%$ Rank-1 scores and $9.8/7.5 \times 10^{-2}s$ query time per probe image. When using 2048 binary codes, its Rank-1 scores increase to $81.4\%/82.5\%$ with query time slow down to $2.8/2.0 \times 10^{-1}s$. This observation is also verified with our method CtF (with AiO) using different code lengths. Importantly, our method CtF+OSF significantly outperforms all existing hashing ReID methods in terms of both accuracy and speed ($5\times$ faster). Specifically, CtF with AiO achieves high accuracy very close to AiO without CtF using 2048 code length, but yields significant speed advantage that is comparable to much shorter 128 binary code length. Besides, OSF further speeds up by around $2\times$ with almost no accuracy drop. Finally, powered by CtF and OSF, our proposed method outperforms state-of-the-art hashing ReID method SIAMH by $1.2\%/2.0\%$ mAP scores and $11.7\times/10\times$ faster querying speed on Market-1501/DukeMTMC-reID datasets.

4.5 Evaluation on Stronger Baselines

A typical person ReID model includes three modules, i.e. *backbone* (*e.g.* ResNet [47], ViTB16 [68]) to extract feature maps from images, *neck* (PCB [11], MGN [60]) to refine features and *head* (*e.g.* Triplet [10], IDE [2]) to train those features. For example, BoT [23], one of the popular non-hashing ReID methods, utilises the ResNet-50 backbone to extract feature maps, a global average pooling (GAP)

TABLE 4
Comparisons with non-hashing ReID methods on Market-1501 and DukeTMTC-reID. Existing non-hashing ReID methods can be grouped into global features, local features, stronger backbones and better pre-training. If not specified, the ImageNet [22] pre-trained ResNet-50 [47] backbone as default. $\mathbf{B}$: binary code, $\mathbf{R}$: real-value feature. Longer real-value features usually have higher accuracy but slower query speed. Our CtF+OSF has very fast query speed (two orders of magnitude faster) and comparable accuracy with non-hashing ReID methods. Besides, CtF+OSF is scalable to stronger baselines such as LUPNL (better pre-training) [70], VITB16 (stronger backbone) [68] and MGN (fine-grained features) [60].

Methods	Year	Code		Market-1501			DukeMTMC-reID		
		Type	Length	R1(%)	mAP(%)	***Q.Time***(s)	R1(%)	mAP(%)	***Q.Time***(s)
global features									
PSE [50]	CVPR'18	$\mathbf{R}$	1,536	78.7	56.0	-	-	-	-
PN-GAN [51]	ECCV'18	$\mathbf{R}$	1,024	89.4	72.6	-	73.6	53.2	-
IDE [52]	Arxiv'16	$\mathbf{R}$	2,048	88.1	72.8	-	69.4	55.4	-
Camstyle [53]	CVPR'18	$\mathbf{R}$	2,048	88.1	68.7	-	75.3	53.5	-
MHN-6(IDE) [54]	ICCV'19	$\mathbf{R}$	2,048	93.6	83.6	-	87.5	75.2	-
SFT(IDE) [55]	ICCV'19	$\mathbf{R}$	2,048	93.4	82.7	-	86.9	73.2	-
BoT [23]	CVPRW'19	$\mathbf{R}$	2,048	94.1	85.7	2.2×10^{0}	86.4	76.4	2.0×10^{0}
M^3 [56]	CVPR'20	$\mathbf{R}$	2,048	95.4	82.6	-	84.5	68.5	-
KPM&GSRW [57]	TPAMI'21	$\mathbf{R}$	2,048	93.7	86.8	-	83.4	71.3	-
GoogleNet+PGR [52]	TPAMI'22	$\mathbf{R}$	2,048	93.8	77.2	-	83.4	71.3	-
OQGFF [58]	ICTAT'21	$\mathbf{R}$	2,048	95.8	88.5	-	90.4	78.3	-
IS-GAN$_{DC}$ [59]	TPAMI'21	$\mathbf{R}$	2,048	96.1	89.4	-	90.8	80.3	-
fine-grained features									
MGN [60]	MM'18	$\mathbf{R}$	2,048	95.1	87.5	-	89.0	79.4	-
SCSN(3stages) [56]	CVPR'20	$\mathbf{R}$	2,304	95.7	88.5	-	90.1	79.0	-
HONet [29]	CVPR'20	$\mathbf{R}$	4,096	94.2	84.9	-	86.9	75.6	-
SPReID [61]	CVPR'18	$\mathbf{R}$	10,240	92.5	81.3	-	84.4	71.0	-
PCB [11]	ECCV'18	$\mathbf{R}$	12,288	93.8	81.6	6.9×10^{0}	83.3	69.2	6.3×10^{0}
PCB-U+RPP [62]	TPAMI'21	$\mathbf{R}$	12,288	94.0	84.8	-	85.9	76.4	-
MHN-6(PCB) [54]	ICCV'19	$\mathbf{R}$	12,288	95.1	85.0	-	89.1	77.2	-
VPM [63]	ICCV'19	$\mathbf{R}$	14,336	93.0	80.8	-	83.6	72.6	-
CGFE+FGFE [64]	CVPR'21	$\mathbf{R}$	14,336	94.8	87.7	-	87.4	74.9	-
RANGEv2 [65]	PR'22	$\mathbf{R}$	14,336	94.7	86.8	-	87.0	78.2	-
stronger backbones									
OSNet [66]	TPAMI'22	$\mathbf{R}$	512	94.8	86.7	9.8×10^{-1}	88.7	76.6	7.5×10^{-1}
ViT [67]	CVPR'22	$\mathbf{R}$	2,048	95.0	86.3	-	89.4	78.0	-
ViTB16 [68]	ICCV'21	$\mathbf{R}$	3,840	95.2	89.5	-	90.7	82.6	-
better pre-training									
LUPerson [69]	CVPR'21	$\mathbf{R}$	2,048	96.3	91.0	-	91.0	82.1	-
LUPersonNL [70]	CVPR'22	$\mathbf{R}$	2,048	96.6	91.9	2.2×10^{0}	92.0	84.3	2.0×10^{0}
CFS+ViTB16 [71]	ArXiv'21	$\mathbf{R}$	7,680	96.0	93.3	-	-	-	-
PASS+ViTB16 [72]	ECCV'22	$\mathbf{R}$	7,680	96.7	93.2	-	-	-	-
Ours (BoT as baseline)									
BoT+CtF	Ours	$\mathbf{B}$	mixed	93.7	84.0	4.6×10^{-2}	87.6	74.8	3.7×10^{-2}
BoT+CtF+OSF	Ours	$\mathbf{B}$	mixed	93.7	83.9	2.4×10^{-2}	87.4	74.5	2.0×10^{-2}
Ours (stronger baselines)									
ViTB16+BoT+CtF+OSF	Ours	$\mathbf{B}$	mixed	94.7	86.7	2.4×10^{-2}	90.0	79.5	2.0×10^{-2}
LUPNL+ViTB16+BoT+CtF+OSF	Ours	$\mathbf{B}$	mixed	95.2	89.0	2.4×10^{-2}	91.1	81.2	2.0×10^{-2}
LUPNL+ViTB16+MGN+CtF+OSF	Ours	$\mathbf{B}$	mixed	**95.5**	**90.0**	$\mathbf{2.4 \times 10^{-2}}$	**91.4**	**81.4**	$\mathbf{2.0 \times 10^{-2}}$

TABLE 5
Comparisons with state-of-the-art hashing ReID methods on Market-1501 and DukeTMTC-reID. CtF achieves a good balance between accuracy and speed. OSF furhter speeds CtF up with almost no accuracy drop. CtF and OSF are scalable to stronger baselines such as LUPNL (better pre-training) [70], VITB16 (stronger backbone) [68] and MGN (fine-grained features) [60].

Methods	Years	Code Length	Market-1501			DukeMTMC-reID		
			R1(%)	mAP(%)	**Q.Time**(s)	R1(%)	mAP(%)	**Q.Time**(s)
DRSCH [74]	TIP'15	512	17.1	11.5	-	19.3	13.6	-
DSRH [75]	CVPR'15	512	27.1	17.7	-	25.6	18.6	-
HashNet [76]	ICCV'17	512	29.2	19.1	-	40.8	28.6	-
CSBT [18]	CVPR'17	512	42.9	20.3	-	47.2	33.1	-
PDH [17]	TIP'17	512	44.6	24.3	-	-	-	-
DCH [77]	CVPR'18	512	40.7	20.2	-	57.4	37.3	-
DeepSSH [78]	ICIP'18	512	46.5	24.1	-	-	-	-
ABC [21]	ICME'19	512	69.4	48.5	9.8×10^{-2}	69.9	52.6	7.5×10^{-2}
SSGAH* [37]	ECCV'18	512	89.5	72.7	9.8×10^{-2}	80.0	62.2	7.5×10^{-2}
ABML [38]	TNNLS'21	512	90.6	74.7	9.8×10^{-2}	82.9	65.0	7.5×10^{-2}
DLBC [79]	MM'20	2,048	94.6	87.4	2.8×10^{-1}	88.7	78.5	2.0×10^{-1}
SSR [80]	NC'21	2,048	94.8	86.0	-	88.4	78.6	-
SIAMH [81]	TIP'21	2,048	95.4	88.8	2.8×10^{-1}	90.1	79.4	2.0×10^{-1}
Ours (BoT as baseline)								
BoT+**CtF**	Ours	32 only	60.0	37.7	3.4×10^{-2}	49.5	28.7	2.3×10^{-2}
BoT+**CtF**	Ours	128 only	88.9	71.0	4.2×10^{-2}	78.6	59.4	3.2×10^{-2}
BoT+**CtF**	Ours	512 only	92.8	82.2	9.8×10^{-2}	85.4	71.6	7.5×10^{-2}
BoT+**CtF**	Ours	2,048 only	93.7	85.4	2.8×10^{-1}	87.7	75.7	2.0×10^{-1}
BoT+**CtF**	Ours	mixed	93.7	84.0	4.6×10^{-2}	87.6	74.8	3.7×10^{-2}
BoT+**CtF+OSF**	Ours	mixed	93.7	83.9	2.4×10^{-2}	87.4	74.5	2.0×10^{-2}
Ours (stronger baselines)								
SIAMH+CtF+OSF	Ours	mixed	95.0	88.0	2.4×10^{-2}	89.8	78.6	2.0×10^{-2}
ViTB16+BoT+CtF+OSF	Ours	mixed	94.7	86.7	2.4×10^{-2}	90.0	79.5	2.0×10^{-2}
LUPNL+ViTB16+BoT+CtF+OSF	Ours	mixed	95.2	89.0	2.4×10^{-2}	91.1	81.2	2.0×10^{-2}
LUPNL+ViTB16+MGN+CtF+OSF	Ours	mixed	**95.5**	**90.0**	$\mathbf{2.4 \times 10^{-2}}$	**91.4**	**81.4**	$\mathbf{2.0 \times 10^{-2}}$

neck to get global feature vectors, and IDE/Triplet heads (linear layers with cross-entropy and triplet loss) to train them. PCB uses the ResNet-50 backbone to extract feature maps, a PCB neck to split a feature map to 6 local feature vectors (*e.g.* body, legs, feet), and 6 corresponding IDE heads (linear layers with cross-entropy losses) to train them. Our proposed CtF and OSF are kinds of heads, where the former maps real-value features to binary codes of different lengths and the latter learns latent attributes. Thus, they should be able to be applied to any baseline with different backbones and necks. As shown in Table 4, recent progress on Re-ID methods can be grouped into global features, fine-grained features, stronger backbones and better pre-training. The first two can be viewed as necks and the last two backbones. To validate the scalability of CtF and OSF to backbones and necks, we report metrics on stronger baselines, including a more advanced backbone (VITB16 [68]), a better backbone-pertaining strategy (LUPersonNL [70]) and a better neck (MGN). Powered by the three advanced modules, mAP scores of CtF+OSF on Market-1501/DukeMTMC-reID are improved by 6.1% and 6.9% again. This demonstrates the scalability of Ctf and OSF to backbones and necks.

4.6 Evaluation on Large-Scale ReID dataset

This section evaluates our proposed method on the large-scale dataset MSMT [82]. MSMT includes 4,101 identities and 126,441 images, which is more challenging than Market-1501 and DukeMTMC-reID datasets. Experimental results are shown in Table 6. As we can see, firstly, the performance of both non-hashing and hashing methods on MSMT is much worse than on Market-1501 and DukeMTMC-reID, showing that larger datasets are usually more complicated than smaller ones. For example, the best non-hashing ReID method LUPersonNL gets $93.3\%/84.3\%$ mAP scores on Market-1501/DukeMTMC-reID, but only 66.1% on MSMT. This is because that larger dataset (i.e. more identities) introduces more hard negative samples and leads to frequent false alarms. Secondly, hashing methods perform worse than non-hashing methods and the phenomenon becomes

TABLE 6
Comparisons with state-of-the-art non-hashing and hashing ReID methods on the large-scale dataset MSMT. Results show that our proposed CtF+OSF performs well on large-scale dataset.

Methods	Code Type	R1 (%)	mAP (%)	Q.Time(s)
GoogleNet+PGR [52]	R	63.0	36.9	-
PCB-U+RPP [62]	R	69.8	43.6	-
KPM&GSRW [57]	R	71.8	47.8	-
OSNet [66]	R	79.1	55.1	-
LUPerson [69]	R	85.1	66.1	-
LUPersonNL [70]	R	86.0	68.0	-
ViTB16 [68]	R	86.2	69.4	-
PASS+ViTB16 [72]	R	89.7	74.3	1.1×10^{2}
R101+DLBC [79]	B	78.9	56.4	-
SIAMH [81]	B	83.2	62.5	6.8×10^{-1}
PASS+ViTB16+CtF+OSF (Ours)	mixed	86.5	69.4	7.2×10^{-2}

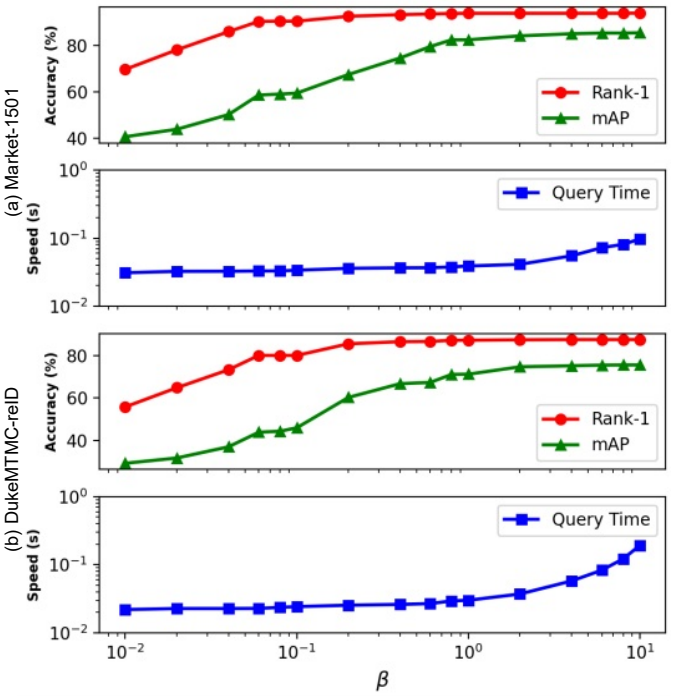

Fig. 6. Analysis of the Distance Threshold Optimization (DTO) module. DTO can well balance accuracy and speed with β. With the increase of β, the accuracy increases and speed becomes slow gradually.

worse on MSMT. For example, the mAP gap on Market-1501 betweeen the best non-hashing and hashing methods (*i.e.* PASS+ViTB16 and SIAMH) are 4.4%. but the mAP gap on MSMT is 11.8%. The reason is that more identities require more fine-grained cues but the quantization process of binary code often loses them. Finally, compared non-hashing methods, our proposed CtF+OSF achieves comparable accuracy with much faster speed. Further, CtF+OSF beats state-of-the-art hashing methods with aspect to both accuracy and speed. This is because CtF (including AiO) utilizes all-in-one module, which enriches cues of binary codes with self-distillation (delivers cues from longer codes to shorter codes), and OSF filters negative samples with a look-up table, which naturally avoids quantization loss.

4.7 CtF Analysis

Analysis of AiO. The All-in-One (AiO) module aims to learn and enhance multiple codes of different lengths in a single model. It uses code pyramid (CP) structure and self-distillation (SD) learning. Results are in Table 7. Firstly, longer codes contribute to better accuracy. This can be seen in all settings no matter whether CP or SD is used and what code type is. Secondly, when using short codes, real-value features is much better than binary ones. But for long codes, they obtain similar accuracy. For example, the 32-dimensional real-value feature obtains 82.7% Rank-1 score, outperforming the 32-dimensional binary code by 60%, where the latter achieved only 25.5%. But when using 2048 code length, binary codes and real-value features both achieve approx. Rank-1 94% and mAP 84%. This suggests that the quantization loss of short codes is significantly worse than that of longer codes. Thirdly, learning with code pyramid (CP) structure or self-distillation (SD) improves short codes significantly. For example, CP+SD boosts the 32-dimensional binary codes from 25.5% to 60.0% in Rank-1 score, upto 35% gain. It is evident that both code pyramid (CP) structure and self-distillation (SD) learning contribute to the effectiveness of the coarse-to-fine (CtF) search strategy, and significantly improve model performance.

Analysis of DTO. We further analyzed parameter β of the Distance Threshold Optimization (DTO) algorithm, which controls the balance between ReID accuracy and speed. Figure 6 show the model accuracy and speed using different β value on Market-1501 and DukeMTMC-reID. Firstly, it is evident that the value of β has a good control of accuracy and speed, increasing β slows down the speed but improves accuracy. For example, when $\beta = 10^{-2}$, ReID is fastest at approx. 0.03 and 0.02 seconds to ReID each probe image on Market-1501 and DukeMTMC-reID, but with mAP scores only at 40% and 30%. In contrast, $\beta = 10^{1}$ gives high mAP 85% and 75%, but the query speed is $5\times$ slower at approx. 0.1 and 0.2 seconds. Secondly, when β is close to 10^{0}, Rank-1 and mAP are almost peaked with a good balance on speed.

Analysis on Larger Gallery. Gallery size affects significantly ReID search accuracy and speed. To show the effectiveness of our proposed Coarse-to-Fine (CtF) search strategy, we evaluated it on a large-scale ReID dataset Market1501+500k. The dataset is based on the Market-1501 and enlarged with $500,000$ distractors. We compare our CtF with three ReID methods, including a non-hashing ReID method with 2048-dimensional real-value features, a hashing ReID model with long binary codes of 2048-dimension, and a hashing ReID model with short binary codes of 32-dimension. The experimental results are shown in Figure 7. We can observe the following phenomenons.

Firstly, with the increase of gallery size, for all methods, the Rank-1 and mAP scores decrease, and the ReID speed per probe image slows down gradually. The reason is that more gallery images is more likely to contain more difficult samples. They make ReID search more challenging. Also, the extra gallery images significantly increase the time

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3340923

TABLE 7
Analysis of the All-in-One (AiO) module. **CP**: learn multiple codes in a pyramid structure, otherwise separate models. **SD**: enhance binary codes via self-distillation. $\mathbf{B}$ and $\mathbf{R}$ mean binary codes and real-value features, respectively.

AiO	CP	SD	Feature Type	Rank-1(%) 32	128	512	2048	CtF	mAP(%) 32	128	512	2048	CtF
×	×	×	**B**										
✓	×	×	**B**	25.5	84.8	92.3	93.8	92.5	33.9	67.5	81.4	85.3	75.1
✓	✓	×	**B**	54.4	87.8	92.7	93.8	93.0	35.0	72.2	81.7	85.3	80.2
✓	✓	✓	**B**	60.0	88.9	92.9	93.8	93.7	37.7	71.0	82.0	85.3	84.0
upper bound			**R**	82.7	90.9	93.4	94.2	-	66.7	78.9	84.3	85.4	-

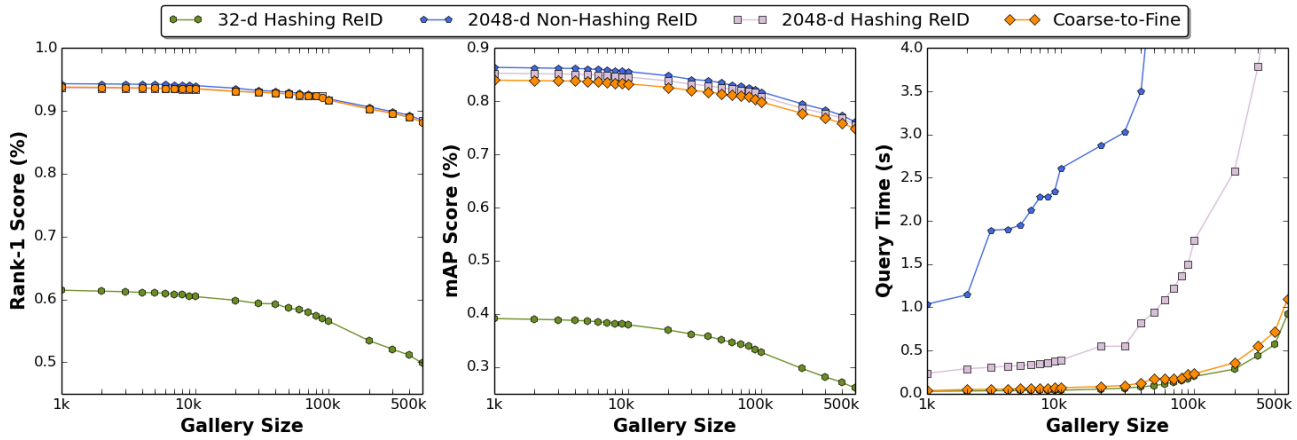

Fig. 7. Experimental results on large-scale ReID dataset Market-1501+500k. Our Coarse-to-Fine (CtF) get a high accuracy comparable with non-hashing ReID method of long code and fast speed comparable with hashing ReID method of short code.

for computing all the distance comparisons and sorting required for ReID each probe image. Secondly, the non-hashing method with 2048-D real-value feature achieves the best accuracy but the worst time. This is because the real-value feature is more discriminative but slow to compute and sort. Thirdly, for hashing ReID methods, the 2048-D binary code obtains comparable ReID accuracy to that of the non-hashing model, but $10\times$ faster. This is because Hamming distances and counting sort are faster to compute. ReID speed of 32-D binary code is $5\times$ faster than that of 2048-D binary codes, but its accuracy drops dramatically. Finally, the proposed CtF model achieves a comparable accuracy to that of the non-hashing method but the advantage of similar speed to that of a hashing ReID method of 32-D binary code. Critically, the advantage is independent of the gallery size. Overall, these experiments demonstrate the effectiveness of CtF for a large-scale ReID task.

Analysis of time and space complexity. We analyse the effect of CtF on time and space complexity. As in TABLE 8, we utilise 4 metrics (3 for time and 2 for space), including FLOPs (float-pointing operations of inference one image), PARAMS (total parameter number of the model), LATENCY (latency of inferring an image) and STORAGE (disk of binary codes per image) and MEM (inference-time memory cost per image). Please note that all metrics above have already considered the backbone module. All metrics are evaluated under ResNet-50 [47] backbone and 128×256 image size using public tools *thop*[2] and *torchstat*[3]. We evaluated LATENCY using the PyTorch backend, without employing ONNX or TRT, on a single 3090 GPU, with a batch size of 256. Experimental results show that the CtF module takes almost no extra inference-time time complexity and no inference-time space complexity. The reason is that the All-in-One (AiO) module includes only three $Linear$ layers and takes $2048 \times 512 + 512 \times 128 + 128 \times 32 = 1M$ FLOPs thus makes no effect to a ResNet-50 backbone which takes 2.7G FLOPs. Only one noticeable metric is STORAGE which takes $1.32\times$ increase compared to the baseline version. But it should be acceptable considering $5\times$ matching-time

TABLE 8
Complexity analysis of the CtF module. The CtF module carries little time and space consuming. Please find details in context.

Metrics	w/o CtF	w/ CtF	Increase
FLOPs	2.70235G	2.70347G	1.0026x
PARAMS	27.61117M	28.7314M	1.0405x
STORAGE	0.25KB	0.332KB	1.32x
MEM	71.65MB	71.66MB	1.00001x
LATENCY	1.8ms	1.9ms	1.056x

2. https://github.com/Lyken17/pytorch-OpCounter
3. https://github.com/Swall0w/torchstat

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3340923

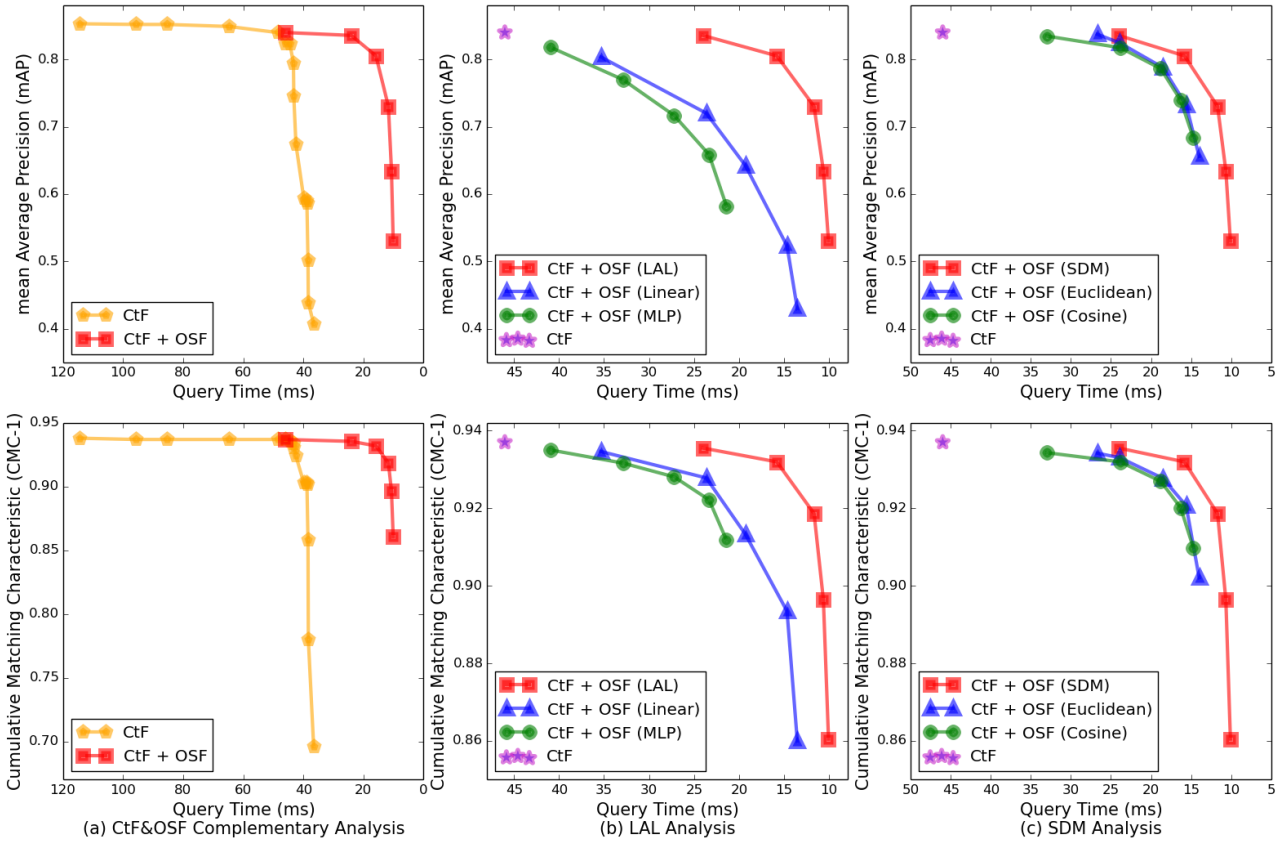

Fig. 8. Analysis of the proposed One-Shot-Filter (OSF) strategy on Market-1501 dataset. (a) OSF is complementary with CtF, further speeding CtF up with less accuracy drop. (b) The proposed Latent-Attribute-Learning (LAL) module is better than common Linear and MLP modules, achieving higher speed with less accuracy drop. (c) The proposed Single-Direction-Metric (SDM) Loss performs better than Euclidean and Cosine metrics.

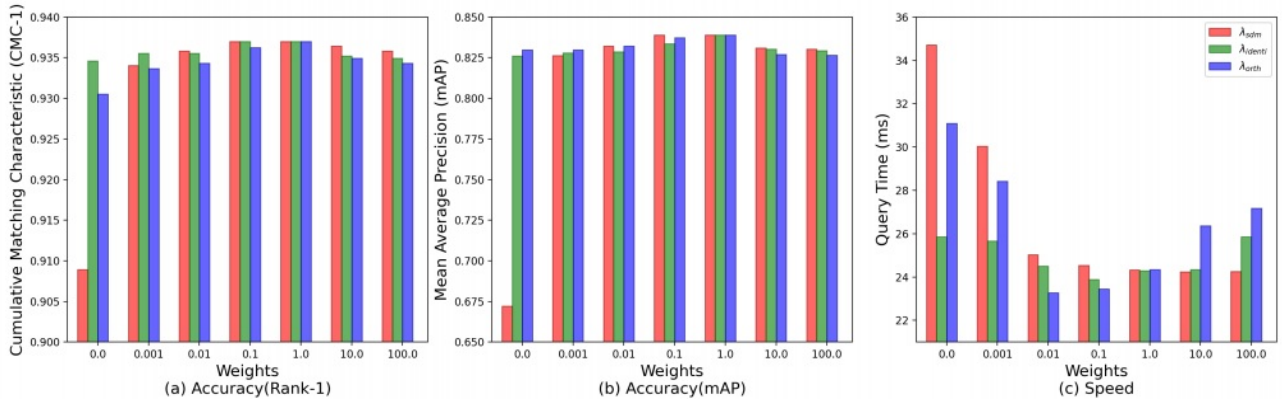

Fig. 9. Parameter analysis of One-Shot-Filter (OSF) strategy. λ_{sdm}, $_{identi}$ and λ_{orth} are weights in Eq.(20). OSF is robust to parameters.

speedup.

4.8 OSF Analysis

OSF is Complementary with CtF. The proposed Coarse-to-Fine (CtF) and One-Shot-Filter (OSF) Strategies are complementary with each other. Relationship between accuracy (mean average precision, mAP) and speed (query time) of Market-1501 dataset is shown in Figure 8(a), where points of CtF (yellow star) are with different parameters β of Eq.(6) (from left to right are10^{-2} to 10^{1}), ones of CtF+OSF (red square) set $\beta = 2.0$ and uses different parameters γ in Algorithm 3 (from left to right are 0, 1, 2, 3, 4 and 5 respectively). As we can see, CtF well balances accuracy and speed with parameter β. Increasing β slows down the speed but improves accuracy. For example, when $\beta = 10^{-2}$, ReID is fastest at approx. 0.03 to ReID each probe image on Market-1501, but with mAP scores only at 40% and 30%. In contrast, $\beta = 10^{1}$ gives high mAP 85%, but the query speed is $5\times$ slower at approx. 0.1 and 0.2 seconds. When β is close to 10^{0}, Rank-1 and mAP are almost peaked with a good balance on speed. Besides, γ has good control of accuracy and speed. Increasing γ slows down the speed but improves accuracy. For example, from $\gamma = 6$, ReID is fastest at about $10ms$ per query image but has low accuracy at 40% mAP and 85% Rank-1. Further, OSF is complementary with OSF. Powered by OSF, CtF ($\beta = 2$)

is speeded up again with less accuracy drop. For example, based on CtF($\beta = 2.0$), increasing γ reduces speed from $46ms$ to about $18ms$ while increasing β only reduce to about $38m$. In summary, CtF+OSF speed up CtF from $46ms$ to $24ms$ with almost no accuracy drop.

Effectiveness of the Latent-Attribute-Learning (LAL) Module. The proposed LAL module is derived from principal component analysis (PCA) that keep the largest variance with the least dimensions, whilst keeping the ability of batch and end-to-end learning. To verify its advantages, we compare LAL with several variants, including a Fully-Connected-Layer (Linear) and Multiple-Layer-Perception (MLP). The experimental results are shown in Figure 8(b). As we can see, all three versions (LAL, Linear and MLP) could achieve faster speed than a single CtF under the acceptable cost of accuracy, demonstrating the effectiveness and robustness of the proposed One-Shot-Filter (OSF) strategy. Among them, our LAL performs best, speeding up $2\times$ with almost no accuracy drop. Specifically, when $\gamma = 1$, CtF+OSF(LAL) speeds up CtF from $46ms$ to $24ms$ with only 0.1% mAP and 0.02% Rank-1, while Linear and MLP get slower speed at $40ms$ and $35ms$ and large accuracy drop by 2% mAP and 0.5% Rank-1.

Analysis of the Single-Direction-Metric (SDM) Loss. SDM loss is specifically designed for the attributed. In this part, we compare SDM with common Euclidean and Cosine metrics. The experimental results are shown in Figure 8(c). As we can see, with either Euclidean, Cosine or SDM, the proposed One-Shot-Filter (OSF) strategy performs well, showing it effectiveness and robustness. Secondly, Cosine and Euclidean metrics own similar performance. For example, CtF+OSF(Euclidean) ($\eta = 1$) speed CtF from $45ms$ to $27ms$ with about 1.0% Rank-1 and 0.5% mAP drop, and CtF+OSF(Cosine) ($\eta = 1$) only get a $34ms$ with similar accuracy. Finally, our proposed CtF+OSF(SDM) has the fastest speed at $24ms$ with less accuracy drop. Experimental results show that the proposed SDM performs better than common Euclidean and Cosine metrics.

Parameter Analysis. In this part, we analyze three parameters that effect OSF, including λ_{sdm}, λ_{identi} and λ_{orth} of Eq.(20). The experimental results are displayed in Figure 9. We can observe several phenomenons. Firstly, non-zero parameters perform better (higher accuracy and faster speed) than zero-parameters), demonstrating that OSF is robust to the three parameters. Secondly, among the three losses ($\mathcal{L}_{sdm}$, L_{identi} and $\mathcal{L}_{eigen}$), $\mathcal{L}_{sdm}$ affects a lot, followed by $\mathcal{L}_{eigen}$ and $\mathcal{L}_{identi}$. Specifically, removing $\mathcal{L}_{sdm}$, *i.e.* setting $\lambda_{sdm} = 1$, significantly reduces Rank-1 and mAP from 93.7% and 83.9% to 91.0% and 68.0%, respectively, and slow speed down from $24ms$ to $35ms$. Removing $\mathcal{L}_{identi}$ and $\mathcal{L}_{eigen}$ only slightly reduce accuracy and slown speed down. Removing $\mathcal{L}_{identi}/\mathcal{L}_{eigen}$ leads to about $0.5\%/0.2\%$ rank-1 and $0.3\%/0.3\%$ mAP drops, and $3ms/11ms$ speed slower.

4.9 Scalability Analysis

Recently, lots of ReID methods are proposed to improve accuracy by polishing real-value features with more advanced backbone architectures, person-related backbone pre-training strategies and more refined local cues. Our proposed One-Shot-Filter (OSF) and Coarse-to-Fine (CtF) Search strategies are based on real-value features and further used to speed up search stage, which is complementary existing real-value ReID methods. In this section, we apply OSF and CtF to three typical real-value methods to show the complementary.

This section valids the effectiveness of our proposed One-Shot-Filter (OSF) and Coarse-to-Fine (CtF) Search strategies under different backbones and larger dataset MSMT [82]. MSMT includes $4,101$ identities and $126,441$ images. The baselines refer TransReID [68] and their codes except for mapping the final layer to $2,048$ dimensions with an extra linear layer for a fair comparison. We analyze two kinds of popular backbones including CNN series (ResNet-50 [47], ResNet-101 [47], ResNet-152 [47], ResNeSt50 [83], ResNeSt200 [83]) and transformer series (DeiT-S/16 [84], DeiT-B/16 [84], ViT-B/16 [84], ViT-B/$16_{s=14}$ [85], ViT-B/$16_{s=12}$ [85]). Details are shown in Table 9.

As we can see, with larger (*e.g.* more layers) and more advanced backbones (e.g. using transformers instead of CNNs), the baseline methods get higher accuracy. Of course, the query times are equal since they all use $2,048$ dimensional real-value features. Powered by our proposed OSF and CtF, the query times are dramatically reduced meanwhile, keeping accuracies comparable.

5 Conclusion

In this work, we proposed novel One-Shot-Filter together with Coarse-to-Fine (CtF) search strategy for faster person re-identification whilst also improving accuracy on conventional hashing ReID. OSF upgrade retrieval-by-ranking to retrieval-by-indexing, which filters easy negative samples by attribute matching. OSF include two key components which are Latent-Attribute-Learning (LAL) module and Single-Direction-Metric Loss. The former learns latent attributes without explicit attribute annotation. The latter optimizes latent attributes in an IOU-like metric, which performs better than common Euclidean and Cosine metrics. CtF first coarsely ranks a gallery using shorter binary codes, then iteratively utilises longer binary codes to further refine on ranking selected top candidates with increasing accuracy. To implement the CtF strategy, a novel All-in-One (AiO) module together with a Distance Threshold Optimization (DTO) algorithm are formulated. The former simultaneously learns and enhances multiple binary codes of different lengths in a single model. The latter solves the complex parameter search task by a simple optimization process. The balance between search accuracy and speed is easily controlled by a single parameter. Extensive experiments show that our method is $5\times$ faster than existing hashing ReID methods but achieves comparable accuracy with non-hashing ReID models that are $50\times$ slower. Based on CtF, OSF further speeds querying up by $2\times$ with almost no accuracy drop.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (62202041), Shenzhen General Research Project under Grant JCYJ20220531093215035,

TABLE 9
Scalability analysis on different backbones and larger dataset MSMT [82]. Our proposed method works well and stably under different backbones, which significantly speed searching up meanwhile keep comparable accuracy.

Backbones	Backbone Complexity	Baseline			OSF+CtF (Ours)		
		MAP(%)	Rank-1(%)	**Q.Time(s)**	MAP(%)	Rank-1 (%)	**Q.Time(s)**
ResNet50	$1\times$	51.2	75.1	2.8×10^1	49.7	74.8	1.2×10^{-1}
ResNet101	$1.48\times$	54.0	76.8	2.8×10^1	51.9	76.4	1.2×10^{-1}
ResNet152	$1.96\times$	55.2	78.0	2.8×10^1	53.3	77.5	1.2×10^{-1}
ResNeSt50	$1.86\times$	61.3	82.0	2.8×10^1	59.4	81.8	1.2×10^{-1}
ResNeSt500	$3.12\times$	63.1	83.5	2.8×10^1	61.1	83.3	1.2×10^{-1}
DeiT-S/16	$0.97\times$	55.0	76.4	2.8×10^1	53.5	76.0	1.2×10^{-1}
DeiT-B/16	$1.79\times$	61.5	81.9	2.8×10^1	59.9	81.7	1.2×10^{-1}
ViT-B/16	$1.79\times$	61.3	81.8	2.8×10^1	59.9	81.6	1.2×10^{-1}
ViT-B/$16_{s=16}$	$2.14\times$	63.7	82.7	2.8×10^1	62.0	82.4	1.2×10^{-1}
ViT-B/$16_{s=12}$	$2.81\times$	64.4	83.5	2.8×10^1	62.8	83.1	1.2×10^{-1}

Fundamental Research Funds for the Central Universities (2023JBMC057), and China Scholarship Council (201904910606).

REFERENCES

[1] S. Gong, M. Cristani, S. Yan, and C. C. Loy, *Person Re-Identification*, 2014.

[2] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[3] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. C. H. Hoi, "Deep learning for person re-identification: A survey and outlook," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2872–2893, 2022.

[4] B. Ma, Y. Su, and F. Jurie, "Covariance descriptor based on bio-inspired features for person re-identification and face verification," *Image and Vision Computing*, vol. 32, no. 6-7, pp. 379–390, 2014.

[5] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *European conference on computer vision*. Springer, 2014, pp. 536–551.

[6] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2197–2206.

[7] W.-S. Zheng, S. Gong, and T. Xiang, "Reidentification by relative distance comparison," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 653–668, 2013.

[8] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 2288–2295.

[9] S. Liao and S. Z. Li, "Efficient psd constrained asymmetric metric learning for person re-identification," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3685–3693.

[10] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification." *arXiv preprint arXiv:1703.07737*, 2017.

[11] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. 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)*, 2018, pp. 480–496.

[12] C. A. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962.

[13] K. Bajpai and A. Kots, "Implementing and analyzing an efficient version of counting sort (e-counting sort)," *International Journal of Computer Applications*, vol. 98, no. 9, 2014.

[14] J. Chen, Y. Wang, and R. Wu, "Person re-identification by distance metric learning to discrete hashing," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 789–793.

[15] F. Zheng and L. Shao, "Learning cross-view binary identities for fast person re-identification," in *IJCAI'16 Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016, pp. 2399–2406.

[16] L. Wu, Y. Wang, Z. Ge, Q. Hu, and X. Li, "Structured deep hashing with convolutional neural networks for fast person re-identification," *Computer Vision and Image Understanding*, vol. 167, pp. 63–73, 2017.

[17] F. Zhu, X. Kong, L. Zheng, H. Fu, and Q. Tian, "Part-based deep hashing for large-scale person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4806–4817, 2017.

[18] J. Chen, Y. Wang, J. Qin, L. Liu, and L. Shao, "Fast person re-identification via cross-camera semantic binary transformation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5330–5339.

[19] W. Fang, H.-M. Hu, Z. Hu, S. Liao, and B. Li, "Perceptual hash-based feature description for person re-identification," *Neurocomputing*, vol. 272, no. 1, pp. 520–531, 2018.

[20] X. Zhu, B. Wu, D. Huang, and W.-S. Zheng, "Fast open-world person re-identification," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2286–2300, 2018.

[21] Z. Liu, J. Qin, A. Li, Y. Wang, and L. V. Gool, "Adversarial binary coding for efficient person re-identification," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 700–705.

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[23] H. Luo, Y. Gu, X. Liao, S. Lai, and W. 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*, 2019, pp. 0–0.

[24] G. Wang, T. Zhang, J. Cheng, S. Liu, Y. Yang, and Z. Hou, "Rgb-infrared cross-modality person re-identification via joint pixel and feature alignment," in *In Proceedings of the 2019 IEEEInternational Conference on Computer Vision (ICCV)*, 2019, pp. 3622–3631.

[25] G. Wang, T. Zhang, Y. Yang, J. Cheng, J. Chang, and Z. Hou, "Cross-modality paired-images generation for rgb-infrared person re-identification," in *In Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[26] G. Wang, Y. Yang, J. Cheng, J. Wang, and Z. Hou, "Color-sensitive person re-identification," in *IJCAI'19 Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 933–939.

[27] G. Wang, S. Gong, J. Cheng, and Z. Hou, "Faster person re-identification." *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[28] G. Wang, Y. Yang, T. Zhang, J. Cheng, Z. Hou, P. Tiwari, H. M. Pandey *et al.*, "Cross-modality paired-images generation and aug-

mentation for rgb-infrared person re-identification," *Neural Networks*, vol. 128, pp. 294–304, 2020.

[29] G. Wang, S. Yang, H. Liu, Z. Wang, Y. Yang, S. Wang, G. Yu, E. Zhou, and J. Sun, "High-order information matters: Learning relation and topology for occluded person re-identification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Scg ?4: Proceedings of the Twentieth Symposium on Computational Geometry*, 2004, pp. 253–262.

[31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *International Conference on Neural Information Processing Systems*, 2008, pp. 1753–1760.

[32] S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 817–824.

[33] W. Liu, J. Wang, R. Ji, and Y. G. Jiang, "Supervised hashing with kernels," in *Computer Vision and Pattern Recognition*, 2012, pp. 2074–2081.

[34] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 37–45, 2015.

[35] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," 2014.

[36] W. J. Li, S. Wang, and W. C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *International Joint Conference on Artificial Intelligence*, 2016, pp. 1711–1717.

[37] G. Wang, Q. Hu, J. Cheng, and Z. Hou, "Semi-supervised generative adversarial hashing for image retrieval," in *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 469–485.

[38] G. Wang, Q. Hu, Y. Yang, J. Cheng, and Z.-G. Hou, "Adversarial binary mutual learning for semi-supervised deep hashing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

[39] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[40] F. Tung and G. Mori, "Similarity-preserving knowledge distillation." in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1365–1374.

[41] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *European Conference on Information Retrieval*. Springer, 2005, pp. 345–359.

[42] A. Balsubramani, S. Dasgupta, and Y. Freund, "The fast convergence of incremental pca," *Advances in neural information processing systems*, vol. 26, 2013.

[43] M. Jia, X. Cheng, Y. Zhai, S. Lu, S. Ma, Y. Tian, and J. Zhang, "Matching on sets: Conquer occluded person re-identification without alignment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 2, 2021, pp. 1673–1681.

[44] M. Jia, X. Cheng, S. Lu, and J. Zhang, "Learning disentangled representation implicitly via transformer for occluded person re-identification," *IEEE Transactions on Multimedia*, vol. 25, pp. 1294–1305, 2022.

[45] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1116–1124.

[46] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," *arXiv preprint arXiv:1701.07717*, 2017. [Online]. Available: https://academic.microsoft.com/paper/2949257576

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[48] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *arXiv preprint arXiv:1708.04896*, 2017.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[50] M. S. Sarfraz, A. Schumann, A. Eberle, and R. Stiefelhagen, "A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 420–429.

[51] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue, "Pose-normalized image generation for person re-identification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 661–678.

[52] J. Li, S. Zhang, Q. Tian, M. Wang, and W. Gao, "Pose-guided representation learning for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 2, pp. 622–635, 2022.

[53] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera style adaptation for person re-identification," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5157–5166. [Online]. Available: https://academic.microsoft.com/paper/2963289251

[54] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[55] C. Luo, Y. Chen, N. Wang, and Z. Zhang, "Spectral feature transformation for person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[56] J. Zhou, B. Su, and Y. Wu, "Online joint multi-metric adaptation from frequent sharing-subset mining for person re-identification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[57] Y. Shen, T. Xiao, S. Yi, D. Chen, X. Wang, and H. Li, "Person re-identification with deep kronecker-product matching and group-shuffling random walk," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, pp. 1–1, 11 2019.

[58] L. Zhang, N. Jiang, Q. Diao, D. Huang, Z. Zhou, and W. Wu, "Object quality guided feature fusion for person re-identification," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021, pp. 1083–1087.

[59] C. Eom, W. Lee, G. Lee, and B. Ham, "Is-gan:learning disentangled representation for robust person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[60] G. Wang, Y. Yuan, X. Chen, J. Li, and X. Zhou, "Learning discriminative features with multiple granularities for person re-identification," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 274–282.

[61] M. M. Kalayeh, E. Basaran, M. Gökmen, M. E. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1062–1071.

[62] Y. Sun, L. Zheng, Y. Li, Y. Yang, Q. Tian, and S. Wang, "Learning part-based convolutional features for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 902–917, 2021.

[63] Y. Sun, Q. Xu, Y. Li, C. Zhang, Y. Li, S. Wang, and J. Sun, "Perceive where to focus: Learning visibility-aware part-level features for partial person re-identification." pp. 393–402, 2019.

[64] A. Zhang, Y. Gao, Y. Niu, W. Liu, and Y. Zhou, "Coarse-to-fine person re-identification with auxiliary-domain classification and second-order information bottleneck," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 598–607.

[65] G. Wu, X. Zhu, and S. Gong, "Learning hybrid ranking representation for person re-identification," *Pattern Recognition*, vol. 121, p. 108239, 2022.

[66] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 5056–5069, 2022.

[67] Z. Wang, F. Zhu, S. Tang, R. Zhao, L. He, and J. Song, "Feature erasing and diffusion network for occluded person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4754–4763.

[68] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "Transreid: Transformer-based object re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 013–15 022.

[69] D. Fu, D. Chen, J. Bao, H. Yang, L. Yuan, L. Zhang, H. Li, and D. Chen, "Unsupervised pre-training for person re-identification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 750–14 759.

[70] D. Fu, D. Chen, H. Yang, J. Bao, L. Yuan, L. Zhang, H. Li, F. Wen, and D. Chen, "Large-scale pre-training for person re-identification with noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2476–2486.

[71] K. Zhu, H. Guo, T. Yan, Y. Zhu, J. Wang, and M. Tang, "Part-aware self-supervised pre-training for person re-identification," *arXiv preprint arXiv:2203.03931*, 2022.

[72] H. Luo, P. Wang, Y. Xu, F. Ding, Y. Zhou, F. Wang, H. Li, and R. Jin, "Self-supervised pre-training for transformer-based person re-identification," *arXiv preprint arXiv:2111.12084*, 2021.

[73] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose-invariant embedding for deep person re-identification," *IEEE Transactions on Image Processing*, vol. 28, no. 9, pp. 4500–4509, 2019.

[74] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 24, no. 12, p. 4766, 2015.

[75] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1556–1564.

[76] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5609–5618.

[77] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for hamming space retrieval," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1229–1237.

[78] Y. Zhao, S. Luo, Y. Yang, and M. Song, "Deepssh: Deep semantic structured hashing for explainable person re-identification," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 1653–1657.

[79] J. Chen, J. Qin, Y. Yan, L. Huang, L. Liu, F. Zhu, and L. Shao, "Deep local binary coding for person re-identification by delving into the details," in *Proceedings of the 28th ACM International Conference on Multimedia*, ser. MM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3034–3043. [Online]. Available: https://doi.org/10.1145/3394171.3413979

[80] H. Jin, S. Lai, G. Zhao, and X. Qian, "Hashing person re-id with self-distilling smooth relaxation," *Neurocomputing*, vol. 455, pp. 111–124, 2021.

[81] C. Zhao, Y. Tu, Z. Lai, F. Shen, H. T. Shen, and D. Miao, "Salience-guided iterative asymmetric mutual hashing for fast person re-identification," *Trans. Img. Proc.*, vol. 30, p. 7776–7789, jan 2021. [Online]. Available: https://doi.org/10.1109/TIP.2021.3109508

[82] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 79–88.

[83] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.

[84] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[85] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

Guan'an Wang received his Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CASIA), China, in 2021. Before that, he received his BS degree from the Central South University (CSU), China, in 2015, and visited to Queen Mary University of London (advised by Prof. Shaogang Gong), UK, from 2019 to 2020, under the foundation of CSC. His research interests are in computer vision, pattern recognition and person re-identification. He has published more than 20 papers in international journals and conferences, including CVPR, ICCV, ECCV, AAAI, IJCAI, TNNLS and so on. He received the 2020 Neural Networks Best Paper Award and 2022 Nomination for Outstanding Doctoral Dissertation of China Society of Image and Graphics (CSIG).

Xiaowen Huang received her BS degree from Central South University in 2015, and her Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (CASIA) in 2020. She is currently a lecturer at the School of Computer and Information Technology, Beijing Jiaotong University. Her research interest is in multimedia analysis and data mining. Her research works are published in the top international conferences and journals, including ACM Multimedia, CVPR, TKDE, TOIS, TOMM, and so on.

Shaogang (Sean) Gong is Professor of Visual Computation at Queen Mary University of London and a Turing Fellow of the Alan Turing Institute of Data Science and Artificial Intelligence. He established the Queen Mary Computer Vision Laboratory in 1993 and has enjoyed immensely working with PhD students and postdoctoral researchers. His research is in Computer Vision and Machine Learning, with a focus on Object Recognition, Action Recognition, and Video Analysis.

Jian Zhang received the B.S. degree from the Department of Mathematics, Harbin Institute of Technology (HIT), Harbin, China, in 2007, and received his M.Eng. and Ph.D. degrees from the School of Computer Science and Technology, HIT, in 2009 and 2014, respectively. From 2014 to 2018, he worked as a postdoctoral researcher at Peking University (PKU) and King Abdullah University of Science and Technology (KAUST). Currently, he is an Assistant Professor with the School of Electronic and Computer Engineering, Shenzhen Graduate School, Peking University, Shenzhen, China. His research interests include low-level vision, AI-generated content (AIGC) and security, and 3D scene understanding. He has published over 100 technical articles in refereed international journals and proceedings. He received the Best Paper Award at the 2011 IEEE Visual Communications and Image Processing (VCIP) and was a co-recipient of the Best Paper Award of 2018 IEEE MultiMedia.

Wen Gao received the Ph.D.degree in electronics engineering from The University of Tokyo, Japan, in 1991. He was a Professor of computer science with the Harbin Institute of Technology, from 1991 to 1995, and a Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He is currently a Professor of computer science with Peking University, China. He has published extensively including five books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as the IEEE ICME and the ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations. He served or serves on the Editorial Board for several journals, such as the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Multimedia, the IEEE Transactions on Image Processing, the IEEE Transactions on Autonomous Mental Development, the EURASIP Journal of Image Communications, and the Journal of Visual Communication and Image Representation.