ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
Received: 21 June 2025; Accepted: 23 September 2025

https://doi.org/10.17559/TV-20250621002765
Original scientific paper

Domain-Robust Deep Hashing: A Unified Framework for Fast Person Re-Identification

Qi LUO

Abstract: Person re-identification (ReID) in large-scale surveillance requires methods that are both accurate and efficient. While deep hashing enables compact binary representations, it often suffers from accuracy degradation due to the domain gap between raw features and hash codes. This paper proposes a unified, open-source framework for fast person ReID that introduces a cross-domain loss function to explicitly bridge the feature and hash spaces. Our model-agnostic training strategy integrates seamlessly with existing architectures such as ResNet and OSNet. Experiments on Market1501 and CUHK03 demonstrate that the proposed framework outperforms state-of-the-art deep hashing and fast ReID methods, achieving up to 8.61% higher mean Average Precision (mAP). Extensive ablation studies validate the contribution of the cross-domain loss, and evaluations across multiple backbones confirm the framework's versatility. The results show that our approach not only improves accuracy but also provides a strong, reproducible baseline for efficient person re-identification.

Keywords: deep hashing; domain gap; fast person re-identification; metric learning

1 INTRODUCTION

Person re-identification (ReID) [1, 24-29] is a task of searching a given probe person from a large collection of images across multiple cameras. In real application scenarios, with the popularity of terminal devices such as cameras, the amount of data that servers need to process increases sharply. How to reduce storage pressure and improve image retrieval speed has become a hot issue in both academia and industry. To tackle this challenge, recently, hashing [2, 3, 16] has been introduced into the ReID community because it is very efficient in terms of computation and storage. The goal of hashing is to embed high-dimensional raw features into a Hamming space of low-dimensional binary hash codes, so that the hash codes of the similar objects are as close as possible, and the hash codes of dissimilar objects are as different as possible. Although deep hashing methods have emerged in succession in recent years, the accuracy of hash coding is still much lower than the raw features in retrieval.

The term "domain gap" [4] is often mentioned in many studies related to domain generalization. Some researchers believe that the existence of domain gap is because the model is not domain-agnostic, which leads to the unique feature distribution for each domain, eventually the performance of the model is greatly reduced when testing on unseen domains. We believe that something similar happens during the process of transforming raw features F into hash codes H.

With the above motivation, we propose a new framework for fast person re-identification. The new baseline consists of three main parts. The first is a feature extractor that extracts the global feature G of pedestrians from RGB images. It can be a convolutional neural network or a transformer [1, 28, 29]. The second is a hash coding module, which maps the global feature G to specific dimensions and obtains the raw feature F and the hash code H. Consequently, with the above two parts, the forward propagation of features can be realized. For training, in addition to the widely used triplet loss and cross entropy loss, we design a cross-domain loss to optimize all of the outputs (F and H) of the hash module at the same time. Specifically, the cross-domain loss function can eliminate the effect of domain gap. After training, the distribution of F and H obtained by our framework is consistent on the whole.

The main contributions of our work are summarized as follows:

1) Novel Analysis Perspective for Fast ReID Performance Degradation. We provide a different perspective to analyze the reasons for the significant performance degradation in fast person re-identification (fast ReID). Unlike previous works that focus primarily on network architecture optimization or loss function design, we identify and analyze the fundamental issue of domain gap between raw features and hash codes through visualization analysis.

2) Comprehensive Framework with Novel Cross-Domain Loss. We propose an open-source framework as a strong baseline for fast ReID that addresses the identified domain gap issue. Our framework consists of three main components: a feature extractor for global feature extraction, a hash coding module for feature-to-hash transformation, and a concise training strategy. Besides the conventional triplet loss and cross-entropy loss, we design a novel cross-domain loss function specifically to optimize both raw features F and hash codes H simultaneously. This cross-domain loss effectively eliminates the domain gap by enforcing consistency between the distributions of raw features and hash codes, ensuring that similar identities remain close in both feature spaces while dissimilar identities are well-separated. The loss function improves the robustness of the end-to-end training process and enables better preservation of discriminative information during hash code generation.

3) Extensive Experimental Validation and Model-Agnostic Design. We conduct extensive experiments on two widely used datasets (Market1501 and CUHKNP) to validate the effectiveness of our approach. Our experimental evaluation includes comprehensive comparisons with both state-of-the-art deep hashing methods and existing fast ReID methods. The results demonstrate that our framework is model-agnostic while consistently achieving state-of-the-art retrieval performance on standard benchmarks. Furthermore, we provide detailed ablation studies to analyze the contribution of each component.

2 RELATED WORK

2.1 Deep Hashing Methods

Deep hashing methods have gained significant attention in computer vision due to their ability to generate compact binary representations while preserving semantic similarity. DPSH [14] introduced a deep pairwise-supervised hashing approach that performs simultaneous feature learning and hash-code learning for applications with pairwise labels, achieving state-of-the-art performance in image retrieval applications. HashNet [13] proposed a novel deep architecture that uses continuation method with convergence guarantees to learn exactly binary hash codes from imbalanced similarity data, addressing the ill-posed gradient problem in optimizing deep networks with non-smooth binary activations.

DPN [7] designed a deep polarized network that employs a differentiable bit-wise hinge-like polarization loss to push each channel output far away from zero, bypassing the requirement for pairwise labels while strictly bounding the pairwise Hamming distance based losses. OrthoCos [8] simplified deep hashing by proposing a single learning objective that maximizes cosine similarity between continuous codes and their corresponding binary orthogonal codes, ensuring both hash code discriminativeness and quantization error minimization while achieving code balancing through Batch Normalization.

CSQ [9] introduced a global similarity metric called central similarity, where hash codes of similar data pairs approach a common center while dissimilar pairs converge to different centers, using hash centers constructed via Hadamard matrix and Bernoulli distributions. JMLH [10] proposed an embarrassingly simple approach with only two additional fully-connected layers and a simple classification objective, lower-bounding the Information Bottleneck between data samples and their semantics. GreedyHash [11] adopted the greedy principle to tackle the NP-hard discrete optimization problem by iteratively updating the network toward probable optimal discrete solutions. SDHC [12] developed a supervised deep hashing method that learns multiple hierarchical non-linear transformations with constraints on loss minimization, bit balance, and bit independence.

While these methods have shown promising results in general image retrieval tasks, they often suffer from accuracy degradation when applied to person re-identification due to the unique challenges in this domain, particularly the domain gap between raw features and hash codes.

2.2 Fast Person Re-identification Methods

Fast person re-identification methods specifically target the efficiency challenges in ReID applications while attempting to maintain accuracy. CPDH [15] proposed a Consistency-Preserving Deep Hashing framework that bridges the gap between high-dimensional features and low-dimensional binary vectors by focusing on consistency preservation. The method introduces a noise consistency cost to improve robustness and a topology consistency cost to maintain ordinal relations between high-dimensional feature space and Hamming space.

DMIH [16] introduced a Deep Multi-Index Hashing approach that seamlessly integrates multi-index hashing and multi-branch based networks into the same framework. The method proposes a block-wise multi-index hashing table construction approach and a search-aware multi-index (SAMI) loss to improve search efficiency while maintaining accuracy.

However, existing fast ReID methods are still limited in number and often lack comprehensive open-source implementations, making it difficult for researchers to reproduce results and build upon existing work. Moreover, these methods frequently neglect the fundamental issue of domain gap between raw features and hash codes, leading to suboptimal performance in real-world applications. The challenge remains in achieving both high efficiency and high accuracy simultaneously, as most existing approaches still suffer from accuracy degradation when converting to binary representations.

3 METHODOLOGY

3.1 Notation

We use lowercase letters like w to denote vectors and uppercase letters like W to denote matrices. For an integer N, we use w_N to denote the set $\{w_1, w_2, \ldots, w_N\}$. $\|w\|_2$ denotes the L_2-norm for w. $[\cdot]_+$ is defined as $[x]_+ = \max\{0, x\}$. sign$(\cdot)$ is an element-wise sign function where sign$(x) = +1$ if $x \geq 0$ else sign$(x) = -1$. Furthermore, $\|b_i - b_j\|_c$ denotes the cosine distance between b_i and b_j, i.e.,

$$\left\|b_i - b_j\right\|_C = 1 - \frac{b_i^T b_j}{\left\|b_i\right\|\left\|b_j\right|}.$$

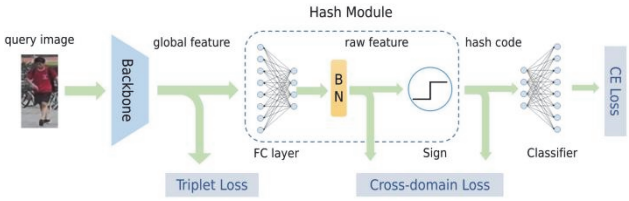


Figure 1 The pipeline of our strong baseline. The green arrowsindicate the flow of features. The part framed by the dottedline is the hash module, which takes global features as inputand outputs raw features and hash codes. The gray rectanglerepresents loss function

3.2 Problem Formulation

Person re-identification with hashing faces several fundamental challenges that make it particularly difficult to optimize end-to-end. The primary challenge lies in the non-differentiable nature of the sign function used to generate binary hash codes.

Let us formally define the problem. Given a set oftraining images $X = \{x_1, x_2, \ldots, x_N\}$ with corresponding identity labels $Y = \{y_1, y_2, \ldots, y_N\}$, our goal is to learn a hash function h: $R^d \rightarrow \{-1, +1\}^k$ that maps high-dimensional features to k-bit binary codes whilepreserving semantic similarity.

3.3 Network Architecture and Forward Propagation

As shown in Fig. 1, let θ denote the parameters of our entire network. For an input image x_i, the forward propagation process can be formulated as follows:

Step 1: Feature Extraction. The input image $x_i \in R^{H \times W \times 3}$ is fed into a backbone network $f_{\text{backbone}}(\cdot; \theta_{\text{backbone}})$ to extract global features:

$$g_i = f_{\text{backbone}}(x_i; \theta_{\text{backbone}}) \tag{1}$$

where $g_i \in R^{dg}$ represents the global feature vector.

Step 2: Hash Module Processing. The global feature g_i is then processed by the hash module consisting of a fully connected layer and batch normalization:

$$f_i = BN\left(FC(g_i; \theta_{\text{hash}})\right) \tag{2}$$

where $f_i \in R^d$ is the continuous raw feature, θ_{hash} represents the parameters of the hash module.

Step 3: Binarization**.** The binary hash code is obtained by applying the sign function:

$$h_i = \text{sign}(f_i) \tag{3}$$

where $h_i \in \{-1, +1\}^d$ is the final binary hash code.

3.4 Challenges

3.4.1 Gradient Vanishing Problem

The core difficulty arises from the sign function sign(·) used to binarize continuous features. Mathematically, for a continuous feature $f \in R^d$, the binary hash code is obtained as:

$$h = \text{sign}\left(f\right) = \begin{cases} +1 & \text{if } f \geq 0 \\ -1 & \text{if } f < 0 \end{cases} \tag{4}$$

The derivative of the sign function is:

$$\frac{\partial \text{sign}\left(f\right)}{\partial f} = \begin{cases} 0 & \text{if } f \neq 0 \\ \text{undefined} & \text{if } f = 0 \end{cases} \tag{5}$$

This zero gradient prevents error signals from propagating back to the backbone network during backpropagation, making it impossible to train the entire network end-to-end.

The challenge in training lies in the backpropagation through the non-differentiable sign function. Given a loss function L, the gradient with respect to raw features becomes:

$$\frac{\partial L}{\partial f_i} = \frac{\partial L}{\partial h_i} \cdot \frac{\partial h_i}{\partial f_i} = \frac{\partial L}{\partial h_i} \cdot \frac{\partial \text{sign}\left(f_i\right)}{\partial f_i} = 0 \tag{6}$$

This zero gradient prevents further backpropagation to the backbone network.

3.4.2 Domain Gap between Continuous and Discrete Representations

To address the gradient vanishing problem mentioned in the previous section, one common approach is to replace the non-differentiable sign function with continuous and differentiable approximations such as the tanh function:

$$h_i = \tanh(\beta \cdot f_i) = \frac{e^{\beta \cdot f_i} - e^{-\beta \cdot f_i}}{e^{\beta \cdot f_i} + e^{-\beta \cdot f_i}} \tag{7}$$

where β is a scaling parameter that controls the steepness of the approximation. While this approach enables gradient flow throughout the network, it introduces several new challenges:

The tanh function produces continuous values in the range (−1, 1), which are not truly binary. During inference, these continuous values must be binarized using the sign function, creating a train-test discrepancy. The model is trained on continuous outputs but tested on discrete ones, leading to a mismatch between training and inference phases. In other words, the transformation from continuous features to discrete hash codes creates a significant domain gap, where the distributions of continuous features and discrete hash codes become inconsistent, leading to suboptimal retrieval performance.

In addition, as β increases to make $\tanh(\beta \cdot f_i)$ more similar to the sign function, the gradients become increasingly small in the saturated regions (where $|\beta \cdot f_i|$ is large), potentially causing training instability and slow convergence.

Besides, the use of approximation functions often requires careful hyperparameter tuning (such as the scaling factor β) and may necessitate complex annealing strategies during training to gradually transition from continuous to discrete representations.

These challenges motivate our approach to directly address the domain gap between continuous features and discrete hash codes through a novel cross-domain loss function, which maintains consistency between the two representations without relying on potentially problematic approximation functions.

3.5 Learning

The framework is shown in Fig. 1, which is an end-to-end deep learning framework containing two components, i.e., main network part and hash codes learning part.

3.5.1 Main Part

Initially, we input a batch of images into the backbone network, and get a batch of global features which is denoted as $\{g\}_N$. Following the setting in [6], for each pedestrian image in a mini-batch, we treat it as an anchor and build the corresponding triplet input by selecting the furthest positive sample and the closest negative sample within the same batch. Thus, the triplet loss $L_t(\{g\}_N)$ can be formulated as follows:

$$L_t\left(\{g\}_N\right) = \frac{1}{N}\sum_{i=1}^{N}\left[\alpha + \max\left\|g_i - g_i^+\right\|_C - \min\left\|g_i - g_i^-\right\|_C\right]_+ \tag{8}$$

where g_i, g_i^+, g_i^- respectively represent the global features from anchor, positive and negative samples, α is the margin hyper-parameter with 0.3 as the default value.

3.5.2 Hash Module Part

As shown in Fig. 1, our hash module consists of three components: 1) a fully connected layer; 2) a batch normalization layer; 3) a sign function. The fully connected layer and batch normalization layer will map $\{g\}_N$ into d-dimensional raw features $\{f\}_N$, where d is a hyper-parameter. In addition, to get the binary code $\{h\}_N$, you need to use the sign function on $\{f\}_N$. According to the previous conjecture, there is a phenomenon of domain gap between the raw feature and the hash code, so in order to make the distribution of the two as consistent as possible, we adopt a cross-domain loss function as shown in Eq. (9):

$$L_{\text{domain}} = \frac{1}{N}\sum_{i=1}^{N}\left[\alpha + \max \| f_i - h_i^+ \|_C - \min \| f_i - h_i^- \|_C\right]_+ \tag{9}$$

where f_i, h_i^+, h_i^- respectively represent the raw features from anchor and the hash codes from its positive and negative samples. $\left\|f_i - h_i^+\right\|_C$ represents the distance between the raw feature and the positive hash code, while $\left\|f_i - h_i^-\right\|_C$ represents the distance between the raw feature and the negative hash code. Eq. (9) encourages alignment between raw features and their hash codes, preventing collapse across domains. Like Eq. (8), the default value of α is also 0.3. In this case, the distance from the raw feature of anchor to the hash code of its positive sample is minimized, while the distance from the raw feature of anchor to the hash code of its negative sample is maximized. Notably, different from the multi-stage method in [2], our framework integrates the high-dimensional raw feature learning and the binary hash code learning into one stage. Additionally, the cross-domain loss directly optimizes the distance between the raw feature and the hash code, instead of maintaining their topology consistency [2]. The distance from the raw feature of anchor to the hash code of its positive sample is minimized, while the distance from the raw feature of anchor to the hash code of its negative sample is maximized.

3.5.3 Loss Functions

Like various deep ReID methods [6], we use the cross entropy loss $L_{ce}(\{h\}^N)$ and label smoothing for classification.The cross entropy loss is defined as:

$$L_{ce}\left(\{h\}^N\right) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(p_{i,c}\right) \tag{10}$$

where N is the batch size, C is the number of identity classes, $y_{i,c}$ is the ground truth label (1 if sample i belongs to class c, 0 otherwise), and $p_{i,c}$ is the predicted probability that sample i belongs to class c. The predicted probabilities are obtained by applying a softmax function to the hash codes:

$$p_{i,c} = \frac{\exp\left(W_c^T h_i + b_c\right)}{\sum_{j=1}^{C}\exp\left(W_j^T h_i + b_j\right)} \tag{11}$$

where W_c and b_c are the weight vector and bias term for class c respectively, and h_i is the hash code of sample i.

To improve generalization and prevent overfitting, we employ label smoothing regularization, which replaces the hard target labels with a mixture of the ground truth and a uniform distribution:

$$y_{i,c}^{\text{smooth}} = (1-\varepsilon)\, y_{i,c} + \frac{\varepsilon}{C} \tag{12}$$

where ε is the smoothing parameter, typically set to 0.1. This technique helps the model to be less confident about its predictions and reduces overfitting to the training data.

The cross entropy loss serves as a classification objective that encourages the hash codes to be discriminative for different person identities, while the cross-domain loss ensures consistency between the continuous and discrete feature representations. The detailed calculation formula will not be introduced here due to the limited space. The overall loss is the sum of the loss in each part:

$$L = L_{\text{base}} + L_{\text{domain}} \tag{13}$$

where $L_{\text{base}} = L_t(\{g\}_N) + L_{ce}(\{h\}_N)$. Unlike other methods [1-3] that add a lot of complicated optimizations, our method only needs to add a cross-domain loss function L_{domain} based on the commonly used L_{base}.

3.6 Testing

In conventional ReID tasks, raw features and the cosine distance are usually used during testing. Conversely, hash codes and the Hamming distance are highly recommended for efficiency-first tasks. To make the experiment more comprehensive, we will show the performance of our framework in these two retrieval strategies in the ablation experiment.

3.7 Discussion

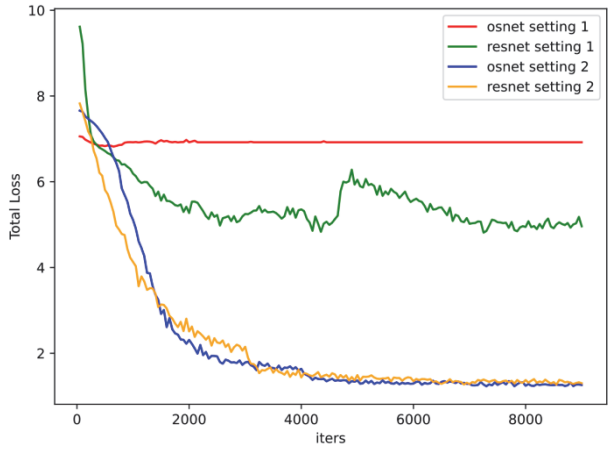
It is obvious thatboth the raw feature and the hash code can represent pedestrian information well, so it is reasonable to use the raw feature of anchor to retrieve the hash code of gallery set, as well as to use the hash code of anchor to retrieve the raw feature of gallery set. Thus, different from previous ReID works, we add two new retrieval strategies in the ablation experiment.

Compared with the Topology Consistency Loss [15] inCPDH, which focuses on preserving the ordinal relations of samples between the feature space and the Hamming space, our proposed cross-domain loss directly enforces

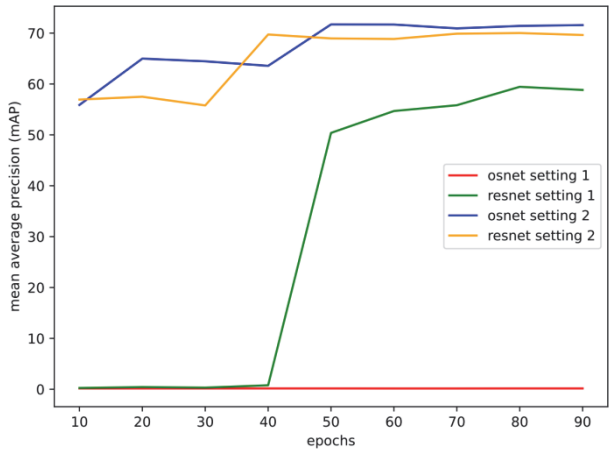
alignment between raw features and hash codes across positive and negative pairs. This makes optimization more straightforward and ensures stronger discriminability of the learned codes. While topology consistency emphasizes global ranking preservation, our cross-domain loss provides a more explicit and effective supervision signal, yielding hash codes that are both compact and more robust for retrieval.

4 RESULTS

Since there are few open-source fast ReID methods, both deep hashing methods and fast ReID methods are adopted as baselines in order to make the results comprehensive. The state-of-the-art deep hashing methods for comparison include: DPN [7], OrthoCos [8], CSQ [9], JMLH [10],GreedyHash [11], SDHC [12], HashNet [13] and DPSH [14]. The state-of-the-art fast ReID methods for comparison include: CPDH [15] and DMIH [16].



(a) Loss-iteration curves



(b) mAP-epoch curves

Figure 2 Ablation results on Market1501 with binary code length being 128 bits

4.1 Experimental Details

4.1.1 Datasets

Experiments are conducted on three widely used datasets: Market1501 [5], CUHK03 [17] and DukeMTMC-ReID [30]. Specifically, CUHK03, denoted as CUHK-NP, contains 14097 images of 1467 pedestrians captured by 2 camera views. To be more challenging, we adopt the labeled images for evaluation as well as the widely recognized new protocol proposed in. Following the standard evaluation protocol on ReID tasks, the evaluation metrics are the mean Average Precision (mAP) and Cumulated Matching Characteristic (CMC).

4.1.2 Implementation Details

It is worth noting that only settings of the deep hashing methods [7-14] and our framework are introduced here, and settings of the fast ReID methods [15, 16] are consistent with their own papers. For fair comparison, all experiments use ResNet-50 [18] pretrained by ImageNet dataset as the backbone network by default. Input images are resized to 256 × 128 and the same data augmentation as in [6] are used. The batch size is 256, which is made up of 64 pedestrians and 4 images for each pedestrian. Hamming distance is used by default when testing. For all deep hashing methods, other hyper-parameters are set by following the suggestion of their own papers. For our framework, we train the model with the Adam strategy for 90 epochs and the weight decay is set as 0.0005. Further, the initial learning rate is set to 3.5×10^{-4} and divided by 10 at the 40th and 70th epochs respectively. We use a 10-epoch linear warm-up. All experiments are implemented with PyTorch on an NVIDIA RTX 4090 GPU.

4.2 Ablation Study and Qualitative Results

4.2.1 Comparisons with the State-Of-The-Art Methods

Existing state-of-the-art methods can be categorized into two types including the deep hashing methods and hashing-based fast ReID methods. From Tab. 1 and Tab. 2, we can see that our framework outperforms all baselines including deep hashing based ReID methods and deep hashing methods in all cases. Particularly, our method substantially outperforms the latest state-of-the-art method Consistency Preserving Deep Hashing (CPDH) in all cases, and on the more challenging CUHK-NP task, our 64-bit, 128-bit and 512-bit hash codes achieve 4.75%, 7.52% and 8.61% higher on the mean average precision (mAP), respectively.

Table 1 mAP (in percent) on Market1501 dataset. The best results are highlighted with bold

Methods	Market1501			
	32 bits	64 bits	128 bits	512 bits
DPN [7]	8.02	23.08	40.78	56.79
OrthoCos [8]	6.69	19.10	39.68	59.66
CSQ [9]	9.92	23.27	39.60	48.80
JMLH [10]	11.03	24.50	40.69	49.07
GreedyHash [11]	18.56	35.80	47.41	59.56
SDHC [12]	16.35	31.21	45.63	60.19
HashNet [13]	15.25	23.22	28.80	34.87
DPSH [14]	7.90	13.99	20.60	24.92
CPDH [15]	-	58.20	67.20	74.9
DMIH [16]	31.41	49.80	62.24	-
Ours	44.12	60.94	70.44	77.86

Our method outperforms existing models across most metrics on all tasks, as evidenced by the data in the tables. To elucidate why our method surpasses other approaches, it is crucial to highlight the unique integration and functionality of our framework components. The bidirectional image translation network effectively mitigates the domain shift by ensuring style consistency

between the source and target domains, which is pivotal for maintaining the visual coherence of identity features across different environments. Simultaneously, our ReID feature extraction network is finely tuned to distill and enhance identity-specific features that are critical for accurate person re-identification.

Table 2 mAP (in percent) on CUHKNP dataset. The best results are highlighted with bold

Methods	CUHKNP			
	32 bits	64 bits	128 bits	512 bits
DPN [7]	1.47	4.48	10.74	28.96
OrthoCos [8]	1.45	3.92	8.83	27.83
CSQ [9]	1.72	6.38	11.60	23.14
JMLH [10]	2.47	5.02	12.22	23.64
GreedyHash [11]	3.31	10.20	20.77	30.94
SDHC [12]	5.43	11.78	20.81	31.08
HashNet [13]	7.50	9.34	11.25	12.71
DPSH [14]	4.29	10.04	12.97	15.67
CPDH [15]	-	44.10	52.0	58.70
DMIH [16]	24.95	40.28	48.75	-
Ours	34.95	48.85	59.52	67.31

Another noteworthy observation is the variation in performance across different code lengths and datasets. Our method benefits more from longer hash codes (128-bit and 512-bit) because the cross-domain loss fully exploits the increased representational capacity to capture finer identity distinctions, while shorter codes (e.g., 32-bit) are more constrained and cannot preserve enough discriminative information despite our improvements. This explains why the accuracy gain becomes more pronounced as the code length increases. Furthermore, the relative improvement on the CUHKNP dataset is larger than that on Market1501. The reason lies in the greater domain shift and higher intra-class variation present in CUHKNP, which makes traditional methods prone to performance degradation. Our framework, however, explicitly addresses this gap through the integration of feature-hash alignment and cross-domain consistency, allowing it to adapt more effectively in such challenging conditions. Consequently, the performance boost on CUHKNP demonstrates the robustness of our design in handling datasets with complex noise and variability.

4.2.2 Model-Agnostic

In order to prove that the proposed framework is robust to different networks, we embed six different backbone networks into our framework.

Tab. 3 compares the parameter counts of six backbone networks, ranging from the lightweight OSNet x1.0 (2.2M parameters) to the heavy ConvNeXtV2 base (88.7M parameters), demonstrating our framework's model-agnostic capability across different computational budgets. Because ConvNeXtV2 base has a large number of parameters, we have to set its batch size to 64. Other training parameters are the same as those in Tab. 1. In Tab. 4 and Tab. 5, we provide the rank k (k = 1, 5, 10) matching rate and mAP across different networks. The comparison results show that our method could work well regardless of the network used. Notably, our framework can easily achieve state-of-the-art results on the two most widely used networks ResNet-50 [18] and OSNet [21].

Table 3 Model parameters comparison

Models	Parameters ($\times 10^6$)
Resnet50 [18]	25.6
Convenextv2_base [19]	88.7
Densenet121 [20]	7.9
Osnet_x1_0 [21]	2.2
Seresnet50 [9]	28.1
Xception [10]	22.9

Table 4 Evaluation of rank k (k = 1, 5, 10) matching rate (%) and mAP (%) on Market1501 dataset for the influence of backbone networks with 128-bit hash code. The best results are highlighted with bold

Models	Market1501			
	mAP	Rank-1	Rank-5	Rank-10
Resnet50 [18]	70.44	86.07	94.42	96.29
Convenextv2_base [19]	65.08	81.71	93.14	95.52
Densenet121 [20]	65.82	83.14	94.09	96.41
Osnet_x1_0 [21]	71.71	86.05	94.92	97.00
Seresnet50 [22]	65.94	82.10	93.23	95.96
Xception [23]	62.50	80.05	92.40	95.01

Table 5 Evaluation of rank k (k = 1, 5, 10) matching rate (%) and mAP (%) on CUHKNP dataset for the influence of backbone networks with 128-bit hash code. The best results are highlighted with bold

Models	CUHKNP			
	mAP	Rank-1	Rank-5	Rank-10
Resnet50 [18]	59.52	63.86	82.29	89.14
Convenextv2_base [19]	58.82	62.21	82.57	89.43
Densenet121 [20]	52.03	56.93	78.50	87.07
Osnet_x1_0 [21]	61.75	66.50	83.43	88.86
Seresnet50 [22]	53.82	57.14	75.86	84.50
Xception [23]	48.39	52.21	74.86	82.64

Table 6 Evaluation of rank k (k = 1, 5, 10) matching rate (%) and mAP (%) on different retrieval strategies with 128-bit hash code. The cosine distance is used

Strategy	Market1501			
	mAP	Rank-1	Rank-5	Rank-10
Raw-to-raw	80.83	91.92	96.67	97.89
Hash-to-hash	70.44	86.07	94.42	96.29
Raw-to-hash	75.71	88.51	95.72	97.18
Hash-to-raw	76.61	88.78	95.67	96.91

4.2.3 Effect of the Cross-DomainLoss

To verify the effectiveness of the proposed cross-domain loss, we adopt two experimental settings on Market1501 with binary code length being 128 bits. The first setting, denoted as setting 1, is to train the framework without the cross-domain loss. The other setting, denoted as setting 2, is to train the framework with all losses shown in Eq. (13). As shown in Fig. 2, the label "resnet setting 1" indicates that ResNet-50 is used as the backbone network under setting 1. In Fig. 2a, we study the convergence speed of different settings, the average loss values over all mini-batches are computed. As shown in this figure, we can find that if only triplet loss and cross entropy loss are used to train the model, the loss function is difficult to converge. In Fig. 2b, we compute the mean average precision on the test split of Market1501 per 10 epoch during training, and it is clear that the cross-domain loss is crucial to improve the accuracy of the model.

4.2.4 Domain-Agnostic

There are two traditional retrieval strategies: raw-to-raw and hash-to-hash. The raw-to-raw strategy means that the raw features are used for both the query set and the gallery set. The hash-to-hash strategy means that

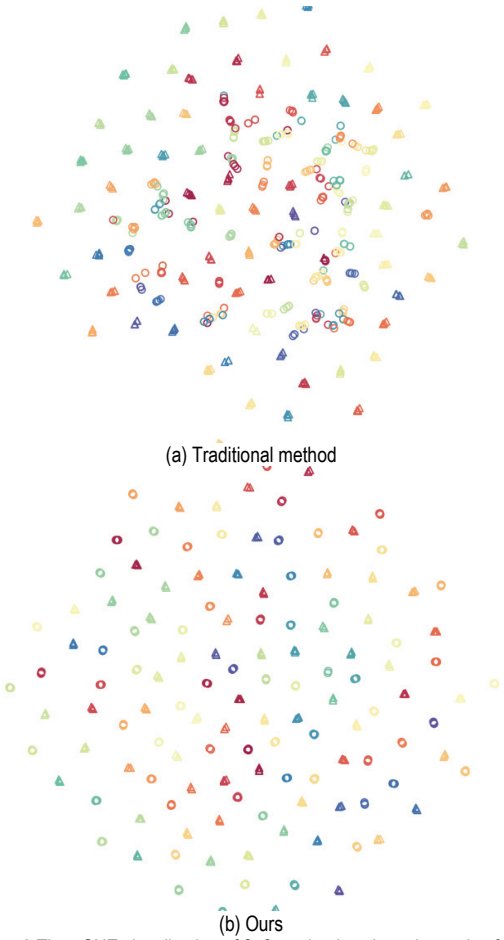
the hash codes are used for both the query set and the gallery set. To test that our method indeed alleviates the domain gap between raw features and hash codes, we add two new retrieval strategies: raw-to-hash and hash-to-raw. Specifically, the raw-to-hash strategy is to use the query sample's raw feature to retrieve hash codes of the gallery set, and the hash-to-raw strategy is to use the query sample's hash code to retrieve raw features of the gallery set. In Tab. 6 and Tab. 7, we apply ResNet-50 as the backbone network and present the performance of our framework across all strategies during retrieving in Market1501 and DukeMTMC-ReID. As shown in these tables, even under the two novel retrieval strategies, our framework can still show high performance.

Table 7 Evaluation of rank *k* (*k* = 1, 5, 10) matching rate (%) and mAP (%) on different retrieval strategies with 128-bit hash code. The cosine distance is used

Strategy	DukeMTMC-ReID			
	mAP	Rank-1	Rank-5	Rank-10
Raw-to-raw	72.82	87.13	92.57	96.41
Hash-to-hash	66.93	78.35	90.20	94.13
Raw-to-hash	70.22	85.37	91.85	96.70
Hash-to-raw	69.05	84.72	91.42	95.16

4.2.5 Visual Analysis

To this end, we used the traditional method to train ResNet-50 on Market-1501 [5], and randomly selected 256 samples for test, including 64 pedestrians, each pedestrian having 4 samples. As shown in Fig. 3a, we used t-SNE to project F and H of these 256 samples into two-dimensional space.



(a) Traditional method

(b) Ours

Figure 3 The t-SNE visualization of 256 randomly selected samples for 64 identities on Market1501. The color represents the identity. Circle means raw features and the triangle means hash codes

Specifically, circular points represent *F*, as well as the triangular points represent *H*, and the same pedestrian is marked with the same color. It can be seen that *F* and *H* follow completely different distributions, and there is an unreasonable phenomenon: the raw features of pedestrian *i* are far away from its own hash codes, but very close to the hash codes of another pedestrian *j*. After training, as shown in Fig. 3b, the strange phenomenon mentioned above disappeared. For example, compared with dissimilar pedestrians, similar pedestrians' raw features and hash codes are much closer.

4.2.6 Trade-offs Between Efficiency and Accuracy

An important consideration of deep hashing-based Re-ID is the balance between accuracy and efficiency. As shown in Tab. 1, our method consistently outperforms prior approaches across all code lengths. The steady increase demonstrates that longer hash codes capture richer discriminative information, leading to improved retrieval accuracy. However, this improvement comes at a computational cost. We measured the average retrieval latency on Market1501 and found that retrieval time increased from (1.28 ± 0.47) ms/query at 64 bits to (6.35 ± 1.4) ms/query at 512 bits, across five independent runs. The results highlight that while longer codes offer higher accuracy, they also significantly increase retrieval time. From a practical standpoint, 128-bit or 256-bit codes provide a favorable trade-off, achieving near-optimal accuracy with manageable overhead.

5 CONCLUSIONS

This study introduced a domain-robust deep hashing framework for fast person re-identification. By addressing the gap between continuous raw features and discrete hash codes through a cross-domain loss, the framework enables more consistent and discriminative binary representations. Experiments on benchmark datasets confirm that the method achieves state-of-the-art performance, with notable gains on challenging CUHK03 tasks and across multiple backbone architectures. The approach proves flexible with different code lengths and retrieval strategies, further validating its robustness. Nevertheless, the work is limited by reliance on only two datasets and by descriptive rather than analytical discussion of results. Future research should extend evaluation to larger and more diverse datasets, incorporate transformer-based backbones, and explore robustness under noise, occlusion, and domain-shift scenarios. With these improvements, the framework has the potential to become a solid baseline for both academic study and real-world large-scale ReID applications.

6 REFERENCES

[1] Zhang, Z., He, D., Liu, S., Xiao, B., & Durrani, T. S. (2023). Completed Part Transformer for Person Re-identification. *IEEE Transactions on Multimedia*, *26*, 2303-2313. https://doi.org/10.1109/TMM.2023.3294816

[2] Li, D., Gong, Y., Cheng, D., Shi, W., Tao, X., & Chang, X. (2019). Consistency-preserving deep hashing for fast person re-identification. *Pattern Recognition*, *94*, 207-217. https://doi.org/10.1016/j.patcog.2019.05.036

[3] Wu, L., Wang, Y., Ge, Z., Hu, Q., & Li, X. (2018). Structured deep hashing with convolutional neural networks for fast person re-identification. *Computer Vision and Image Understanding*, *167*, 63-73. https://doi.org/10.1016/j.cviu.2017.11.009

[4] Wei, L., Zhang, S., Gao, W., & Tian, Q. (2018). Person transfer gan to bridge domain gap for person re-identification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 79-88. https://doi.org/10.1109/CVPR.2018.00016

[5] Zheng, L., Shen, L., Tian, L., Wang, S., & Wang, J. (2015). Scalable person re-identification: A benchmark. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1116-1124. https://doi.org/10.1109/ICCV.2015.133

[6] Luo, H., Gu, Y., Liao, X., Lai, S., & Jiang, W. (2019). Bag of tricks and a strong baseline for deep person re-identification. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 1487-1495. https://doi.org/10.1109/CVPRW.2019.00190

[7] Fan, L., Ng, K. W., Ju, C., Zhang, T., & Chan, C. S. (2020). Deep polarized network for supervised learning of accurate binary hashing codes. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 825-831. https://doi.org/10.24963/ijcai.2020/115

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[9] Yuan, L., Wang, T., Zhang, X., Tay, F. E. H., Jie, Z., Liu, W., & Feng, J. (2020). Central similarity quantization for efficient image and video retrieval. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3083-3092. https://doi.org/10.1109/CVPR42600.2020.00315

[10] Zhou, K., Yang, Y., Cavallaro, A., & Xiang, T. (2021). Learning generalisable omni-scale representations for person re-identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(9), 5056-5069. https://doi.org/10.1109/TPAMI.2021.3069237

[11] Su, S., Zhang, C., Han, K., & Tian, Y. (2018). Greedy hash: Towards fast optimization for accurate hash coding in cnn. *Advances in Neural Information Processing Systems*, *31*.

[12] Liong, V. E., Lu, J., Wang, G., Moulin, P., & Zhou, J. (2015). Deep hashing for compact binary codes learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2475-2483. https://doi.org/10.1109/CVPR.2015.7298862

[13] Cao, Z., Long, M., Wang, J., & Yu, P. S. (2017). Hashnet: Deep learning to hash by continuation. *Proceedings of the IEEE International Conference on Computer Vision*, 5608-5617. https://doi.org/10.1109/ICCV.2017.598

[14] Li, W.-J., Wang, S., & Kang, W.-C. (2016). Feature learning based deep supervised hashing with pairwise labels. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI)*, 1711-1717.

[15] Li, D., Gong, Y., Cheng, D., Shi, W., Tao, X., & Chang, X. (2019). Consistency-preserving deep hashing for fast person re-identification. *Pattern Recognition*, *94*, 207-217. https://doi.org/10.1016/j.patcog.2019.05.036

[16] Li, M.-W., Jiang, Q.-Y., & Li, W.-J. (2019). Deep multi-index hashing for person re-identification. *arXiv preprint arXiv:1905.10980*.

[17] Zhong, Z., Zheng, L., Cao, D., & Li, S. (2017). Re-ranking person re-identification with k-reciprocal encoding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1318-1327. https://doi.org/10.1109/CVPR.2017.389

[18] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778. https://doi.org/10.1109/CVPR.2016.90

[19] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I. S., & Xie, S. (2023). Convnext v2: Co-designing and scaling convnets with masked autoencoders. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16133-16142. https://doi.org/10.1109/CVPR52729.2023.01548

[20] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4700-4708. https://doi.org/10.1109/CVPR.2017.243

[21] Zhang, T., Chan, C. S., Song, Y.-Z., & Xiang, T. (2021). One loss for all: Deep hashing with a single cosine similarity based learning objective. *Advances in Neural Information Processing Systems (NeurIPS)*, *34*, 24286-24298.

[22] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[23] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251-1258. https://doi.org/10.1109/CVPR.2017.195

[24] Zhai, Y. et al. (2020). AD-Cluster: Augmented Discriminative Clustering for Domain Adaptive Person Re-Identification. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9018-9027. https://doi.org/10.1109/CVPR42600.2020.00904

[25] Tang, Q. & Jo, K.-H. (2022). Unsupervised Person Re-identification via Mining Label Homogeneity. *2022 IEEE International Conference on Industrial Technology (ICIT)*, 1-6. https://doi.org/10.1109/ICIT48603.2022.10002807

[26] Yang, F., Li, K., Zhong, Z. et al. (2020). Asymmetric Co-Teaching for Unsupervised Cross-Domain Person Re-Identification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 12597-12604. https://doi.org/10.1609/aaai.v34i07.6950

[27] Zhong, Z., Zheng, L., Luo, Z., Li, S., & Yang, Y. (2021). Learning to Adapt Invariance in Memory for Person Re-Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(8), 2723-2738.

[28] He, S., Luo, H., Wang, P., Wang, F., Li, H., & Jiang, W. (2021). TransReID: Transformer-based Object Re-Identification. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 15013-15022. https://doi.org/10.1109/ICCV48922.2021.01474

[29] Ye, M., Chen, S., Li, C., Zheng, W.-S., Crandall, D., & Du, B. (2025). Transformer for Object Re-identification: A Survey. *International Journal of Computer Vision*, *133*(5), 2410-2440. https://doi.org/10.1007/s11263-024-02284-4

[30] Zheng, Z., Zheng, L., & Yang, Y. (2017). Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in Vitro. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3774-3782. https://doi.org/10.1109/ICCV.2017.405

Contact information:

Qi LUO
Weinan Normal University,
West Section of CHAOYANG Avenue,
Weinan City, Shaanxi Province,China
E-mail: lq_wn@126.com