Central similarity consistency hashing for asymmetric image retrieval

Zhaofeng Xuan[1,2,3], **Dayan Wu**[1](✉), **Wanqian Zhang**[1], **Qinghang Su**[1,2,3], **Bo Li**[1,2,3], **and Weiping Wang**[1,2]

Abstract Asymmetric image retrieval methods have drawn much attention due to their effectiveness in resource-constrained scenarios. They try to learn two models in an asymmetric paradigm, i.e., a small model for the query side and a large model for the gallery. However, we empirically find that the mutual training scheme (learning with each other) will inevitably degrade the performance of the large gallery model, due to the negative effects exerted by the small query one. In this paper, we propose **C**entral **S**imilarity **C**onsistency **H**ashing (CSCH), which simultaneously learns a small query model and a large gallery model in a mutually promoted manner, ensuring both high retrieval accuracy and efficiency on the query side. To achieve this, we first introduce heuristically generated hash centers as the common learning target for both two models. Instead of randomly assigning each hash center to its corresponding category, we introduce the Hungarian algorithm to optimally match each of them by aligning the Hamming similarity of hash centers to the semantic similarity of their classes. Furthermore, we introduce the instance-level consistency loss, which enables the explicit knowledge transfer from the gallery model to the query one, without the sacrifice of gallery performance. Guided by the unified learning of hash centers and the distilled knowledge from gallery model, the query model can be gradually aligned to the Hamming space of the gallery model in a decoupled manner. Extensive experiments demonstrate the superiority of our CSCH method compared with current state-of-the-art deep hashing methods. Code is available here.

1 Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085. Email: Z. Xuan, xuanzhaofeng@iie.ac.cn; D. Wu, wudayan@iie.ac.cn(✉); W. Zhang, zhangwanqian@iie.ac.cn; Q. Su, suqinghang@iie.ac.cn; B. Li, libo@iie.ac.cn; W. Wang, wangweiping@iie.ac.cn.

2 School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China.

3 Key Laboratory of Cyberspace Security Defense.

Corresponding Author: Dayan Wu.

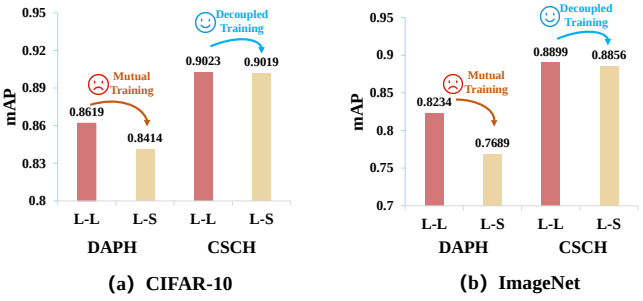


Fig. 1 Retrieval performance of the large gallery model jointly trained with either a large (L-L) or small (L-S) query one. We calculate mAP only using the large gallery model to evaluate the effects of the query model. DAPH exhibits a noticeable decrease in retrieval accuracy due to the mutual training with the small query model. Instead, our CSCH decouples the aforementioned training scheme, and has no negative effects on the gallery model.

Keywords asymmetric image retrieval; deep hashing; Hungarian algorithm; decoupled manner

1 Introduction

Learning to hash has been widely used in several computer vision tasks, e.g., image retrieval [15, 18, 33, 38, 42], video retrieval [28, 44], and person re-identification [36, 43, 45, 48], due to its remarkable efficiency for data storage and retrieval. The key idea is to convert high-dimensional image data into compact binary codes while preserving the semantic similarity between them. Then the retrieval system can calculate the Hamming distance to measure the similarity between the images. Recently, deep hashing methods [15, 23, 25, 29, 33, 38, 39, 42, 44, 46, 49, 50] have greatly improved the performance over traditional hashing methods [11, 12, 37] by harnessing the power of deep learning [5, 13, 16, 21, 47].

In modern image retrieval systems, one practical challenge is to balance the retrieval accuracy and efficiency in resource-constrained scenarios. Most hashing methods [3, 15, 25, 29, 51] usually use the same large model to encode query and gallery images symmetrically for better retrieval performance, named as *symmetric image retrieval* [2]. However, it is inefficient to deploy such a large model on query side

with limited computing resources. Consequently, *asymmetric image retrieval* [2, 40], which indexes the gallery images with a large model while encoding the query images with a small one, is proposed to tackle the balance between the accuracy and efficiency.

Several deep hashing methods [18, 33] have been proposed to asymmetrically learn compatible hashing models. DAPH [33] is the first deep hashing method that introduces two different models for query and gallery images. Specifically, DAPH jointly trains two models to learn pairwise similarity preserving codes in an alternative manner. However, empirical results show that such mutual training scheme will inevitably degrade the performance of the large gallery model. As in Fig. 1, DAPH exhibits a clear retrieval decline on both datasets, due to the negative effects of the small query model. To tackle this, the vanilla knowledge distillation [1, 14, 40], which pre-trains a large gallery model by existing hashing methods and then transfers the knowledge to a small query model, is a simple yet effective solution. However, the performance of the query model heavily relies on that of the gallery one, which will degrade the asymmetric retrieval accuracy once the gallery model fails to learn discriminative hash codes.

In this paper, we propose a novel deep hashing framework, Central Similarity Consistency Hashing (CSCH), which decouples the aforementioned alternative optimization and fully exploits the potential of the two models under the power of central similarity (shown in Fig. 1). First, we design a heuristic algorithm to find the optimal match between hash centers and their corresponding image categories. To avoid the large model being constrained by the small one, each model directly maximizes the cosine similarity between the continuous hash codes and their corresponding hash centers, respectively. This scheme can be considered as the center-level consistency, providing a superior unified learning target for both models. In light of the superior representation capability of the large gallery model, we further introduce an instance-level consistency loss, encouraging the small query model to mimic the large gallery one. As a result, hash codes generated by both models are as similar as possible to the corresponding hash centers. The main contributions are summarized as follows:

- This paper proposes a novel deep hashing framework, named Central Similarity Consistency Hashing (CSCH). To the best of our knowledge, CSCH is the first deep hashing approach for asymmetric image retrieval, which can jointly optimize both the small query model and the large gallery model in an end-to-end manner.
- We introduce the Hungarian algorithm to optimally align the Hamming similarity of hash centers to the semantic similarity of their classes. Furthermore, code consistency loss is proposed to ensure both the center-level and instance-level consistency between hash codes generated by query and gallery models.
- Comprehensive experiments show that our method outperforms the state-of-the-art deep hashing methods by a large margin consistently.

2 Related Work

Asymmetric Image Retrieval. Existing asymmetric image retrieval methods [2, 40, 41] typically involve training a large gallery model. Subsequently, the small query model is optimized using elaborately designed metric losses, all while the pre-trained gallery model remains unchanged. AML[2] proposes an asymmetric metric learning framework that adopts different optimization objectives to encourage the small query model to align with the large gallery model. CSD[40] further introduces a contextual similarity distillation framework which constrains contextual similarity consistency to keep features generated by the query model compatible with that of the gallery model with no labels. AFF[41] enhances current asymmetric retrieval systems by taking into account the complementarity of various features, particularly on the gallery side. However, such the distillation scheme is time-consuming, and the feature quality of the small model largely lies in the large model. Instead, we learn compatible query and gallery models in an end-to-end way while optimizing both models with the unified target to reduce the dependency of small model on large model.

Deep Hashing Methods. Deep hashing methods [15, 18, 25, 33, 42, 44] have shown prominent performance improvements over non-deep hashing methods with hand-crafted features [11, 12, 37]. Recently, central similarity based symmetric deep hashing methods have attracted more attention and presented better performance [10, 15, 35, 44]. CSQ [44] first generates hash centers via Hadamard matrix, and then pulls hash codes towards their corresponding centers with binary cross entropy loss. Similarly, OrthoHash [15] proposes to maximize the cosine similarity between the continuous codes and their corresponding hash centers. Besides, many asymmetric deep hashing methods have been proposed to learn hash codes for query and gallery images with different models. DAPH [33] jointly trains two different models to learn pairwise similarity preserving codes in an alternative manner. However, the larger model is constrained to be aligned with the smaller one. Different from previous asymmetric hashing methods, we leverage central similarity as the unified learning target to train two models in an end-to-end way, which can significantly improve the performance of both models.

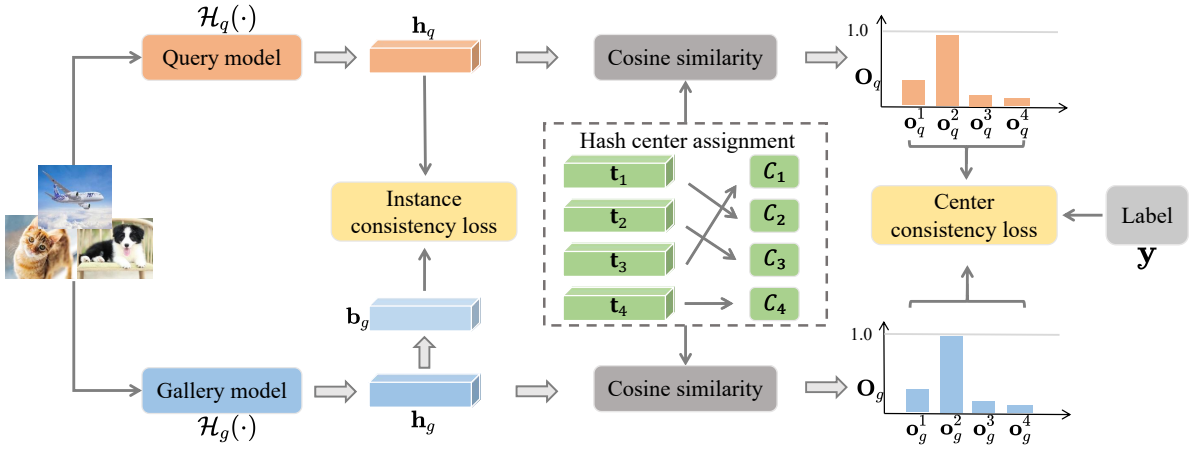


Fig. 2 Pipeline of our CSCH method. Firstly, the two models maximize the cosine similarity between the continuous codes and their corresponding hash centers with center consistency loss respectively. Here, hash centers are designed specifically to find the optimal assignment to classes. Finally, the instance consistency loss directly aligns continuous codes from the query model with binary codes from the gallery model for better asymmetric retrieval performance.

Knowledge Transfer. Knowledge transfer is a widely-adopted model compression technique that aims to transfer knowledge from a large model to a smaller model. A pioneering study by Hinton *et al.* [14] trains a student model with the aim of matching the softmax distribution of a teacher model. RKD[31] transfers mutual relations of data examples, such as distances and angles. Recent researches[2, 40] distill knowledge from the pre-trained teacher model to the student model focusing on various metric losses. However, the performance of the student model heavily depends on the teacher model. Instead of training with the fixed teacher model, we propose an end-to-end training scheme and additionally introduce a superior teacher, i.e., optimally matched hash centers, for both models. By mimicing both the teacher model and the optimal hash centers, the student model is able to generate hash codes compatible with the teacher model, thereby facilitating the asymmetric image retrieval task.

3 The Proposed method

3.1 Preliminaries

Assume that we have a training set of N images $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N \in \mathbb{R}^{N\times D}$, and the corresponding labels $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^N \in \{0,1\}^{N\times C}$, where D is the dimension of images and C is the number of total classes. The goal of deep hashing is to learn a hashing function $\mathcal{H}$: $\mathbf{x} \mapsto \mathbf{b} \in \{-1,1\}^B$ from input space $\mathbb{R}^D$ into Hamming space $\{-1,1\}^B$ via deep networks, where $\mathbf{b} = sgn(\mathbf{h})$ is B-bit binary codes transformed from the continuous codes $\mathbf{h} \in \mathbb{R}^B$ through a sgn function. We pre-define a set of hash centers $\mathbf{T} = [\mathbf{t}_1, \cdots, \mathbf{t}_C]^\top \in \{-1,+1\}^{C\times B}$, where $\mathbf{t}_i$ denotes binary class center belongs to the i-th class. For any two hash centers, $1 \leqslant i, j \leqslant C$, $\mathbf{t}_i$ and $\mathbf{t}_j$ should have sufficient distance in Hamming space, e.g., $D_H(\mathbf{t}_i, \mathbf{t}_j) > d$, where $D_H(\cdot)$ denotes the Hamming distance, and d is the minimum Hamming distance.

Under the asymmetric image retrieval setting, we use a small model $\mathcal{H}_q(\cdot)$ for the query side and a large model $\mathcal{H}_g(\cdot)$ for the gallery side. Given an image $\mathbf{x}$, the hash codes generated from the query / gallery model are denoted as $\mathbf{b}_q = \mathcal{H}_q(\mathbf{x})$ / $\mathbf{b}_g = \mathcal{H}_g(\mathbf{x})$. Our goal is to make the query model $\mathcal{H}_q(\cdot)$ and the gallery model $\mathcal{H}_g(\cdot)$ align with each other. In other words, we aim to encourage $\mathbf{b}_q$ and $\mathbf{b}_g$ to be as similar as possible (ideally $\mathbf{b}_q = \mathbf{b}_g$), while preserving the semantic information behind the images.

3.2 Framework Overview

As illustrated in Fig. 2, the proposed Central Similarity Consistency Hashing (CSCH) framework includes two components: *hash center generation and assignment, code consistency loss*. Powered by central similarity, we first generate hash centers heuristically and assign each hash center to its optimal class label. Then, we adopt two different backbones for both query and gallery sides to learn hash codes in an end-to-end manner. Finally, the code consistency loss ensures the consistency between hash codes generated by query and gallery models, which consists of a center consistency loss and an instance consistency loss. The center consistency loss focuses on preserving the central similarity between query and gallery hash codes with the guidance of the common hash centers. Meanwhile, given each image, the instance consistency loss tries to align the hash codes of the two models, enabling the knowledge transfer from the large gallery model to the small query model.

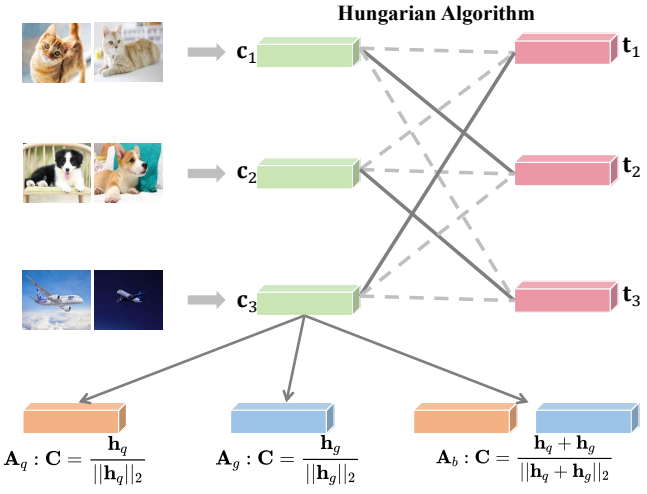


Fig. 3 An illustration of hash center assignment with Hungarian algorithm.

3.3 Hash Center Generation and Assignment

Hash Center Generation. Previous methods [26, 44] usually use the Hadamard matrix to generate strictly orthogonal hash centers, which means the distances between each pair of hash centers are identical. In fact, the semantic distances between classes are not equal as expected, e.g., the semantic distance between a cat and a dog is smaller than that between a cat and an airplane. Thus, it is crucial to acquire hash centers that contain semantic information while ensuring an adequate Hamming distance. Inspired by [15], we attempt to heuristically generate hash centers while guaranteeing that the minimum Hamming distance between hash centers is greater than a specified distance d:

$$\min_{1\leqslant i,j\leqslant C} D_H\left(\mathbf{t}_i, \mathbf{t}_j\right) > d \tag{1}$$

Hash Center Assignment. In [15, 35, 44], the hash centers are randomly assigned to classes. However, this will result in the mismatch between the semantic space of images and Hamming space of hash centers with the same classes, ultimately leading to a degradation of performance. Thus, it is essential to find the optimal assignment between each hash center and its corresponding class.

Figure 3 illustrates the assignment of hash center with Hungarian algorithm. Firstly, we need to accurately measure the semantic space of images. In fact, almost all deep hashing methods deploy a network pre-trained on ImageNet[6] as the initialization. Thus, the continuous codes generated by the initialized network have enough semantic information to quantify the semantic space. Specifically, we feed all training images into the initialized hash network to obtain the normalized class centers $\mathbf{C} = \{\mathbf{c}_i\}_{i=1}^{C}$, where $\mathbf{c}_i$ can be calculated as:

$$\mathbf{c}_i = \frac{\sum_{\mathbf{h}_j \in \mathbf{H}_i} \mathbf{h}_j}{\left\| \sum_{\mathbf{h}_j \in \mathbf{H}_i} \mathbf{h}_j \right\|_2} \tag{2}$$

and $\mathbf{H}_i$ denotes the set of continuous hash codes of images belonging to class i. Under asymmetric retrieval setting, we need to train two models with different backbones for both the query and gallery sides. We conduct sufficient experiments to evaluate the assignment based on the query model ($\mathbf{A}_q$) or the gallery model ($\mathbf{A}_g$) or both of them ($\mathbf{A}_b$).

The next step is to effectively assign hash centers to their respective classes. Here, we adopt the Hungarian algorithm [22, 24] as our assignment algorithm to obtain the optimal match $\Gamma^* = \{\delta_i^*\}_{i=1}^{C}$ between hash centers and normalized class centers:

$$\Gamma^* = \arg\min_{\{\delta_i\}_i} \frac{1}{C} \sum_{i=1}^{C} \|\mathbf{t}_{\delta_i} - \mathbf{c}_i\| \tag{3}$$

Finally, the optimal hash centers $\mathbf{T}^*$ can be assigned as:

$$\mathbf{T}^* = [\mathbf{t}_{\delta_1^*}, \mathbf{t}_{\delta_2^*}, \cdots, \mathbf{t}_{\delta_C^*}]^T \tag{4}$$

3.4 Code Consistency Loss

As shown in Fig. 2, we adopt two different backbones as **deep hash functions** for both the query and gallery sides to learn consistent hash codes. We replace the classification layer in the original neural network with a new hash layer which includes a fully-connected (FC) layer and a batch normalization (BN) layer. Here, the batch normalization layer makes the hash codes more balanced[15]. Thus the deep hash functions for the query and gallery sides are formulated as:

$$\begin{aligned} \mathbf{b}_q &= \mathcal{H}_q(\mathbf{x}) = sgn(f(\mathbf{x}; \theta_q)) \\ \mathbf{b}_g &= \mathcal{H}_g(\mathbf{x}) = sgn(f(\mathbf{x}; \theta_g)) \end{aligned} \tag{5}$$

where θ_* denotes the parameters of the query or gallery model and $f(\cdot)$ denotes the output of the hash layer.

Center-level Consistency. Since Hamming distance between the binary hash codes can be interpreted as cosine similarity [15]. Specifically, for any two continuous hash codes $\mathbf{h}_i$ and $\mathbf{h}_j$ ($1 \leqslant i, j \leqslant N$):

$$D_H\left(\mathbf{b}_i,\ \mathbf{b}_j\right) \simeq \frac{B}{2}\left(1 - \cos\left(\mathbf{h}_i,\ \mathbf{h}_j\right)\right) \tag{6}$$

where $\mathbf{b}_i = sgn(\mathbf{h}_i)$, $\mathbf{b}_j = sgn(\mathbf{h}_j)$. In other words, maximizing the cosine similarity between continuous hash codes is equal to maximizing the Hamming similarity between the binary hash codes. Thus, we calculate the cosine similarities between continuous hash codes $\mathbf{h}_q$, $\mathbf{h}_g$ and the optimal hash centers $\mathbf{T}^*$, to obtain the classification outputs $\mathbf{O}_q$ and $\mathbf{O}_g$, e.g., $\mathbf{O}_q$ can be defined as:

$$\mathbf{O}_q = \left[\cos\left(\mathbf{h}_q, \mathbf{t}_{\delta_1^*}\right), \cos\left(\mathbf{h}_q, \mathbf{t}_{\delta_2^*}\right), \cdots, \cos\left(\mathbf{h}_q, \mathbf{t}_{\delta_C^*}\right)\right] \tag{7}$$

Then, we maximize the cosine similarity of the continuous hash codes $\mathbf{h}_q$ / $\mathbf{h}_g$ and their corresponding hash center $\mathbf{t}_\mathbf{y} \in \mathbf{T}^*$ by minimizing the cross entropy loss between $\mathbf{O}_q$ / $\mathbf{O}_g$ and their corresponding label $\mathbf{y}$, and $\mathcal{L}_q$ can be formulated as:

$$\mathcal{L}_q = -\log \frac{\exp\left(\cos(\mathbf{h}_q, \mathbf{t}_\mathbf{y})\right)}{\sum_{i=1}^{C} \exp\left(\cos(\mathbf{h}_q, \mathbf{t}_{\delta_i^*})\right)} \tag{8}$$

In addition, we utilize scaled cosine similarity with margin [7, 34] to further align $\mathbf{h}_q$ and $\mathbf{h}_g$ with $\mathbf{t}_\mathbf{y}$, encouraging $\mathbf{h}_q$ and $\mathbf{h}_g$ to move closer to each other implicitly. Equation (8) can therefore be rewritten as:

$$\mathcal{L}_q = -\log \frac{\exp\left((\cos(\mathbf{h}_q, \mathbf{t}_\mathbf{y}) - m)/\tau\right)}{\sum_{i=1}^{C} \exp\left((\cos(\mathbf{h}_q, \mathbf{t}_{\delta_i^*}) - m)/\tau\right)} \tag{9}$$

where τ and m are hyper-parameters. And $\mathcal{L}_g$ can be formulated in the same way. Finally, the center consistency loss can be formulated as the sum of $\mathcal{L}_q$ and $\mathcal{L}_g$:

$$\mathcal{L}_C = \mathcal{L}_q + \mathcal{L}_g \tag{10}$$

Instance-level Consistency. Center consistency loss tries to align continuous hash codes $\mathbf{h}_q, \mathbf{h}_g$ with their common hash center $\mathbf{t}_\mathbf{y}$, which aligns $\mathbf{h}_q$ with $\mathbf{h}_g$ implicitly. It is equally essential to explicitly bring the small query model close to the large gallery model to further enhance performance in real-world asymmetric retrieval scenarios. Specifically, in this work, we explicitly constrain the instance-level consistency between the query continuous hash code $\mathbf{h}_q$ and the gallery binary hash code $\mathbf{b}_g = sgn(\mathbf{h}_g)$ with the $\mathcal{L}_2$ distance metric, ensuring the query model is close to the gallery model while keeping the large model from being unintentionally affected by the smaller one. This is reasonable to directly keep $\mathbf{h}_q$ close to $\mathbf{b}_g$ since we adopt $\mathbf{b}_g$ as the gallery code during asymmetric retrieval. Concretely, the instance consistency loss is defined as:

$$\mathcal{L}_I = \|\mathbf{b}_g - \mathbf{h}_q\|_2^2 \tag{11}$$

In a nutshell, by merging Eq. (10) with Eq. (11), we arrive at the definitive formulation of **code consistency loss** for training:

$$\mathcal{L} = \mathcal{L}_C + \gamma \mathcal{L}_I \tag{12}$$

where γ is a hyper-parameter that makes a trade-off between different loss terms.

Multi-Label Hash Codes Learning. As for multi-label images, we also utilize the cross entropy loss to optimize them. Each image, however, belongs to multiple categories, which means one image corresponds to multiple hash centers. Different from previous hashing methods [15, 17] that maximize the similarity between the hash code and its corresponding multiple hash centers, we first obtain the hash centroid $\mathbf{z}$ of the centers like [44] does. Specifically, $\mathbf{z}$ can be calculated as:

$$\mathbf{z} = sgn(\frac{1}{\|\mathbf{y}\|_1} \sum_{i=1}^{C} \mathbf{t}_{y_i}) \tag{13}$$

If the final result is 0 at some bits, we sample from the Bernoulli distribution to set these bits 1 or -1. Then, we maximize the similarity between hash codes and the corresponding hash centroid. Thus, the center consistency loss of the query model $\mathcal{L}_q$ is formulated as:

$$\mathcal{L}_q = -\log \frac{\exp\left(Sim(\mathbf{h}_q, \mathbf{z})\right)}{\exp\left(Sim(\mathbf{h}_q, \mathbf{z})\right) + \sum_{i \in NegC} \exp\left(Sim(\mathbf{h}_q, \mathbf{t}_{\delta_i^*})\right)} \tag{14}$$

where $Sim(\mathbf{h}_q, \mathbf{z}) = (\cos(\mathbf{h}_q, \mathbf{z}) - m)/\tau$ and $NegC$ represents a subset of categories that image $\mathbf{x}$ does not belong to. And $\mathcal{L}_g$ is formulated in the same way. With this learning scheme, both models are able to learn the explicit hash centroid of multi-class labels.

In the end, we summarize the whole learning algorithm for CSCH in Algorithm 1. For further details, please refer to our open-source code.

4 Experiment

4.1 Dataset

We conduct empirical evaluations of our proposed method on three widely used datasets: CIFAR-10, ImageNet and MS-COCO.

CIFAR-10 [20] consists of 60,000 32×32 images in 10 classes. Following [25, 33], we randomly sample 100 images per class as the test set, and the rest are used as the database. Then we randomly select 500 images per class from the database as the training set.

ImageNet [6] contains 1.2M training images and 50K validation images from 1,000 classes. We follow [10, 15] to randomly select 100 classes. Then we use all the images of these classes in the training set as the database and use all the images in the validation set as the test set. Furthermore, we randomly sample 130 images per class from the database as the training set.

MS-COCO [27] contains 82,783 images in the training set and 40,504 images in the validation set. Each image belongs to one of the 80 classes. Following [4], we obtain 122,218 images with category information. Then, we randomly select 5,000 images as the test set, and the rest are used as the database. Finally, we randomly sample 10,000 images from the database as the training set.

Algorithm 1 The learning algorithm for CSCH

Input:

N: number of samples; M: batch size; C: number of classes; θ_*: parameters of hash model; $\mathbf{T}$: randomly generated hash centers; η_*: learning rate; $optimizer_*$: optimizer.

Output:

$\mathbf{T}^*$: optimal hash centers; f_{θ_*}: hash model.

Hash Center Assignment:

compute normalized class centers $\mathbf{C}$ according to (2);
find the optimal match Γ^* between $\mathbf{T}$ and $\mathbf{C}$ with Hungarian algorithm;
$\mathbf{T}^* \leftarrow assign(\Gamma^*, \mathbf{T})$.

Asymmetric Models Optimization:

repeat
  foreach j = 1 to $\frac{N}{M}$ **do**
    $\mathbf{h}_q \leftarrow f(\mathbf{x}_j, \theta_q)$;
    $\mathbf{h}_g \leftarrow f(\mathbf{x}_j, \theta_g)$;
    if $\mathbf{x}_j$ is single-label image **then**
      compute $\mathcal{L}_C$ using $(\mathbf{h}_q, \mathbf{h}_g, \mathbf{T}^*)$ according to (9, 10);
    else
      compute $\mathcal{L}_C$ using $(\mathbf{h}_q, \mathbf{h}_g, \mathbf{T}^*)$ according to (14, 10);
    end if
    compute $\mathcal{L}_I$ using $(\mathbf{h}_q, \mathbf{h}_g)$ according to (11);
    $\mathcal{L} \leftarrow \mathcal{L}_C + \gamma\mathcal{L}_I$;
    $\delta\theta_q \leftarrow \partial_{\theta_q}\mathcal{L}$;
    $\delta\theta_g \leftarrow \partial_{\theta_g}\mathcal{L}$;
    $\theta_q \leftarrow optimizer_q(\theta_q, \delta\theta_q, \eta_q)$;
    $\theta_g \leftarrow optimizer_g(\theta_g, \delta\theta_g, \eta_g)$;
  end foreach
until models converge or reach the max epoches;

4.2 Baselines and Backbone Networks

Baselines. We chose 4 state-of-the-art hashing methods (including DAPH [33], CSQ [44], OrthoHash [15], and MDSH [35]) as our backbone methods. Specifically, DAPH is the first deep hashing method that proposes two different models for both the query and gallery aspects. CSQ, OrthoHash, and MDSH exemplify recent advances in hashing methods, and we compare their performance against CSCH, focusing on hash center assignment strategy and center consistency loss. Besides, under the asymmetric retrieval setting, the last three baselines are modified as two-stage knowledge distillation varieties, which first pre-train a large gallery model using corresponding approaches, and then transfer knowledge to a small query model with $\mathcal{L}_2$ distance metric loss.

Table 1 Comparison of FLOPS and parameter numbers in different backbones.

Query Backbone	Gallery Backbone	GFLOPS		PARAM(M)	
		ABS.	%	ABS.	%
RN50	RN50	4.12	100.0	23.57	100.0
RN101	RN101	7.84	100.0	42.50	100.0
ViT-B/16	ViT-B/16	16.86	100.0	85.67	100.0
MNetv3-S	RN50	0.06	1.46	1.55	6.58
MNetv3-L	RN101	0.23	2.93	4.24	9.98
MViT-XXS	ViT-B/16	1.09	6.47	1.91	2.23

Backbone Networks. We adopt RN50 (ResNet50)[13], RN101 (ResNet101)[13] and ViT-B/16[9] as the large gallery models. MNetv3-S (MobileNetv3-small)[16], MNetv3-L (MobileNetv3-large) [16] and MViT-XXS (MobileViT-XXS)[30] are used as the small query models. Precisely, we conduct experiments with three kinds of combinations, i.e., MNetv3-S and RN50, MNetv3-L and RN101, MViT-XXS and ViT-B/16. All backbone networks are pre-trained on ImageNet. Table 1 shows the computational complexity (in FLOPS) and the number of parameters of different backbones.

4.3 Implementation Details and Metric

Implementation Details. The proposed CSCH is implemented with PyTorch [32]. We train 150 epochs for all three datasets with the batch size of 64. Adam [19] is adopted as the optimizer. The learning rate of MNetv3-S is set to 1e-5 on CIFAR-10, and 2e-5 on ImageNet and MS-COCO respectively, while that of the rest is set to 1e-5 on all datasets. The weight decay is set to 5e-4. We set the hyper-parameter γ to 10 on ImageNet, and 100 on CIFAR-10 and MS-COCO. The hyper-parameter τ is set to 1/8, m is set to 0.3 on CIFAR-10 and 0.2 on ImageNet and MS-COCO. All the experiments are conducted on a PC equipped with an Intel Xeon Silver 4214 CPU@2.20GHz, 128GB RAM, and an NVIDIA RTX 3090 GPU.

Evaluation Metric. We evaluate the asymmetric image retrieval performance of our proposed CSCH and other baseline methods using Mean Average Precision (**mAP**), Precision-Recall curves (**PR curves**) and precision with respect to top-K returned images (**P@Top-**K). Particularly, different from previous symmetric hashing works [4, 10, 15, 25, 44], we encode hash codes of query images utilizing a small query model and obtain hash codes of database images via a larger gallery model to asymmetrically calculate mAP. Equally, we plot PR curves and P@top-K curves employing similar ways.

Table 2 Comparisons of mAP (asymmetric retrieval) with representative hashing methods using CNN-based backbones on CIFAR-10, ImageNet and MS-COCO. Black bold: best results. †: our re-implementation for asymmetric retrieval.

Methods	Query BackBone	Gallery BackBone	CIFAR-10@ALL				ImageNet@1K				MS-COCO@5K			
			12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
Using different backbones for query and gallery models														
DAPH† [33]	MNetv3-S	RN50	0.5970	0.8196	0.8253	0.8539	0.0577	0.4065	0.6591	0.7331	0.6218	0.6921	0.7066	0.7733
CSQ† [44]			0.8262	0.8356	0.8399	0.8481	0.6845	0.7451	0.7574	0.7694	0.6944	0.7926	0.8132	0.8333
OrthoHash† [15]			0.8658	0.8757	0.8841	0.8885	0.6861	0.7468	0.7646	0.7728	0.7441	0.8094	0.8242	0.8449
MDSH† [35]			0.8410	0.8508	0.8460	0.8590	0.6928	0.7557	0.7650	0.7804	0.7036	0.8027	0.8213	0.8450
CSCH			**0.8665**	**0.8792**	**0.8870**	**0.8924**	**0.7291**	**0.7837**	**0.7962**	**0.8073**	**0.7798**	**0.8288**	**0.8440**	**0.8580**
CSCH (Upper Bound)	RN50	RN50	0.8802	0.8964	0.9023	0.9091	0.8572	0.8816	0.8899	0.8982	0.7906	0.8558	0.8671	0.8880
DAPH† [33]	MNetv3-L	RN101	0.6182	0.8545	0.8597	0.8578	0.0681	0.5426	0.7437	0.8106	0.6301	0.6943	0.7326	0.7844
CSQ† [44]			0.8403	0.8437	0.8622	0.8616	0.7690	0.8251	0.8289	0.8418	0.7224	0.8138	0.8233	0.8584
OrthoHash† [15]			0.8841	0.8950	0.9010	0.9038	0.7701	0.8286	0.8370	0.8462	0.7695	0.8389	0.8546	0.8754
MDSH† [35]			0.8594	0.8735	0.8678	0.8749	0.7894	0.8335	0.8417	0.8543	0.7330	0.8321	0.8482	0.8752
CSCH			**0.8890**	**0.8974**	**0.9017**	**0.9057**	**0.8126**	**0.8535**	**0.8604**	**0.8675**	**0.8016**	**0.8557**	**0.8705**	**0.8841**
CSCH (Upper Bound)	RN101	RN101	0.8968	0.9085	0.9206	0.9268	0.8638	0.8986	0.9018	0.9088	0.8039	0.8652	0.8817	0.8969

Table 3 Comparisons of mAP (asymmetric retrieval) with representative hashing methods using Transformer-based backbones on MS-COCO.

Methods	Query Backbone	Gallery Backbone	MS-COCO@5K			
			12 bits	24 bits	32 bits	48 bits
Using different backbones for query and gallery models						
DAPH† [33]	MViT-XXS	ViT-B/16	0.6381	0.7181	0.7217	0.7802
CSQ† [44]			0.7425	0.8149	0.8312	0.8407
OrthoHash† [15]			0.7781	0.8437	0.8573	0.8690
MDSH† [35]			0.7570	0.8323	0.8422	0.8580
CSCH			**0.8079**	**0.8590**	**0.8649**	**0.8751**
CSCH (Upper Bound)	ViT-B/16	ViT-B/16	0.8651	0.9132	0.9235	0.9369

Table 4 Comparisons of mAP with representative asymmetric image retrieval methods and different training strategies. †: our re-implementation.

∇G	$\mathcal{L}_I$	MS-COCO@5K			
		12 bits	24 bits	32 bits	48 bits
-	-	0.7568	0.8112	0.8302	0.8486
✓	AML-Reg†[2]	0.7675	0.8193	0.8296	0.8496
	AML-Contr†[2]	0.6910	0.6986	0.7269	0.7640
✗	AML-Reg†[2]	0.7744	0.8253	0.8399	0.8554
	AML-Contr†[2]	0.6985	0.7009	0.6963	0.7195
	Ours	**0.7798**	**0.8288**	**0.8440**	**0.8580**

4.4 Accuracy Comparison

mAP Comparisons with Representative Hashing Methods. We compare the mAP performance between SOTA deep hashing methods and CSCH on CIFAR-10, ImageNet, and MS-COCO. Table 2 reports the mAP comparisons (symmetric or asymmetric retrieval) of using CNN-based backbones with different lengths of hash code.

Our CSCH outperforms current state-of-the-art by clear margins consistently. Specifically, CSCH surpasses DAPH by significant margins on three datasets (average mAP by 11.7% on CIFAR-10, 31.1% on ImageNet, and 14.2% on MS-COCO) for asymmetric retrieval. When the dataset is relatively simple and easy to learn (like CIFAR-10), both CSCH and OrthoHash achieve high performance. And the retrieval accuracy using both small and large backbones nearly matches the performance achieved when solely employing a large backbone. It emphasizes the importance of a strong teacher model for the small query models. On more sophisticated datasets e.g., single-label dataset ImageNet and multi-label dataset MS-COCO, CSCH achieves superior performance due to the optimal hash centers as additional teachers for the two models. On ImageNet, CSCH outperforms MDSH in average mAP by 2.5%. On MS-COCO, the asymmetric retrieval performance of CSCH is better than OrthoHash in average mAP by 2%. Moreover, it still maintains competitive asymmetric performance when compared to the corresponding larger model only used for symmetric retrieval.

Following the current trend in other computer vision tasks, we report the mAP comparisons of using Transformer-based backbones on MS-COCO, which is shown in Table 3. As reported, Transformer exhibits outstanding performance across all methods consistently. Moreover, our CSCH continues to surpass other methods by substantial margins, resulting in significantly improved retrieval performance and upper bound. This showcases the superiority of our proposed framework.

mAP Comparisons with Representative Asymmetric Image Retrieval Methods. Representative asymmetric image retrieval methods, e.g., AML[2], adopt different instance consistency losses $\mathcal{L}_I$ to encourage the small query model to align with the large gallery model. Table 4 shows the mAP comparisons between SOTA asymmetric image retrieval methods and our CSCH with different training strategies on MS-COCO. The mark ✓ denotes back propagating $\mathcal{L}_I$'s gradients to update the gallery model ∇G, whereas the symbol ✗ signifies no update is needed. AML-Reg and AML-Contr are representative instance consistency losses proposed by AML. To be specific, AML-Reg refers to the enforcement of instance-level consistency between $\mathbf{h}_q$ and $\mathbf{h}_g$ with $\mathcal{L}_2$ distance. On the other hand, AML-Contr employs contrastive loss to constrain $\mathbf{h}_q$ and $\mathbf{h}_g$(s) that share the same label.

Table 5 The mAP comparisons with different combinations of losses.

$\mathcal{L}_q$	$\mathcal{L}_I$	ImageNet@1K				MS-COCO@5K			
		12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
✓		0.7156	0.7764	0.7872	0.8056	0.7568	0.8112	0.8302	0.8486
	✓	0.6939	0.7493	0.7653	0.7791	0.7783	0.8253	0.8417	0.8527
✓	✓	**0.7291**	**0.7837**	**0.7962**	**0.8073**	**0.7798**	**0.8288**	**0.8440**	**0.8580**

Table 6 The mAP results with different strategies of hash center assignment.

Assignment Strategy	ImageNet@1K				MS-COCO@5K			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
$\mathbf{A}_r$	0.7212	0.7780	0.7908	0.8015	0.7556	0.8209	0.8327	0.8538
$\mathbf{A}_q$	0.7291	**0.7837**	**0.7962**	**0.8073**	0.7798	0.8288	**0.8440**	**0.8580**
$\mathbf{A}_g$	**0.7302**	0.7806	0.7922	0.8016	**0.7818**	0.8279	0.8412	0.8557
$\mathbf{A}_b$	0.7263	0.7789	0.7934	0.8050	0.7792	**0.8300**	0.8418	0.8577

The first row in Table 4 reports the results without $\mathcal{L}_I$. We can observe that our proposed $\mathcal{L}_I = \mathcal{L}_2(\mathbf{h}_q, \mathbf{b}_g)$ yields superior results when compared with AML-Reg and AML-Contr. As discussed before, $\mathcal{L}_2(\mathbf{h}_q, \mathbf{b}_g)$ directly constrains $\mathbf{h}_q$ and $\mathbf{b}_g$. This is more appropriate than AML-Reg for asymmetric retrieval, where $\mathbf{b}_g$ serves as the gallery code rather than $\mathbf{h}_g$. As for AML-Contr, much like DAPH, it necessitates that $\mathbf{h}_q$ be implicitly close to $\mathbf{h}_g$(s) with a corresponding label in a mini-batch, and vice versa. Considering that we don't have an adequately large batch size to incorporate a sufficient number of negative samples (i.e., samples with different labels), and our primary aim is just to ensure that the hash code of the image remains consistent across both models. Thus, AML-Contr is unsuitable for instance consistency loss, and even offers negative effects on retrieval performance. Moreover, the training strategies related to the backward propagation of gradients for the gallery model also affect the asymmetric retrieval performance. This confirms our assertion about the negative effects of a small query model impacting a larger gallery one. As a result, our designed instance consistency loss $\mathcal{L}_2(\mathbf{h}_q, \mathbf{b}_g)$ enables the small query model to learn better with the guidance of the large gallery model compared with other asymmetric retrieval methods.

4.5 Ablation Study

Impact of Different Loss Terms. We first evaluate different combinations of losses $\mathcal{L}_q$ and $\mathcal{L}_I$. It should be noted that $\mathcal{L}_g$ is necessary for all empirical settings. Hash center assignment is based on the query model. As shown in Table 5, when removing $\mathcal{L}_I$ (1st row), the performance reduces significantly on MS-COCO for all bits. We summarize the reason as the misalignment between the small query model and the large gallery model. Concretely, the small query model is unable to learn the hash centroid well when compared to the large gallery model on a multi-label dataset. Fortunately, the instance consistency loss can transfer knowledge and assist the small query model in enhancing its learning capabilities. When removing $\mathcal{L}_q$ (2nd row), the mAP decreases consistently for all bits, especially on ImageNet. It demonstrates that the optimal hash centers are beneficial for the query model on the single-label dataset.

Hash Center Assignment Strategy. We assess various strategies for determining the optimal hash centers, which are based on the normalized class centers generated by small query model $\mathbf{A}_q$, large gallery model $\mathbf{A}_g$, and both of them $\mathbf{A}_b$, respectively. Recall that we feed all training images into the initialized hash model to obtain the normalized class centers. Here, the initialized hash model includes a CNN backbone pretrained on ImageNet and a hash layer initialized by a normal distribution with a mean of 0 and a variance of 0.1. The hash layer aims to convert high-dimensional features into low-dimensional hash codes while preserving the semantic similarity among the features. Similar to LSH[11], the hash codes, as the outputs of the initialized hash layer, are able to encompass the semantic similarity present within the features. Besides, the well-matched hash centers, obtained through the Hungarian algorithm with normalized class centers, are also more compatible with the currently initialized hash model(s). Thus, as in Table 6, our proposed assignment method shows superiority compared with random assignment. Specifically, the mAP of $\mathbf{A}_q$ assignment almost achieves the best results compared to the others for all bits. We argue that the performance of asymmetric retrieval heavily relies on that of the small query model, and $\mathbf{A}_q$ assignment can notably enhance the performance of the small query model. This is also in line with our conclusion that randomly assigning hash centers to class labels $\mathbf{A}_r$ will lead to misalignment between Hamming space and semantic space.

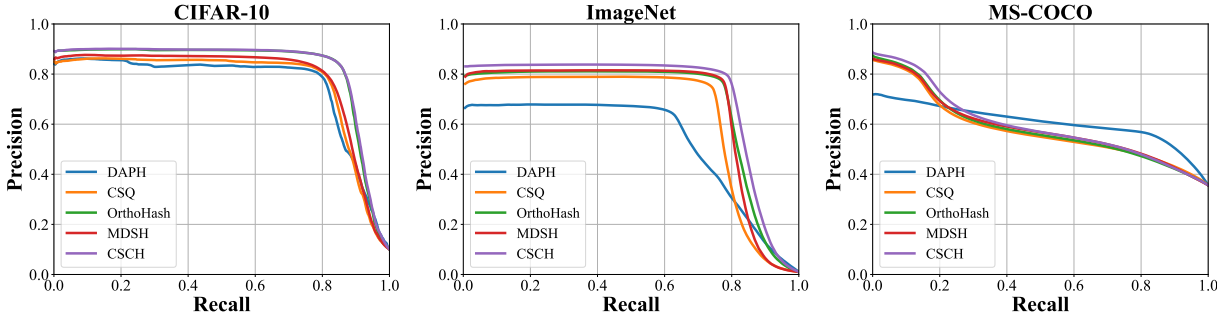


Fig. 4 Precision-Recall curves on CIFAR-10, ImageNet and MS-COCO with 32 bits.

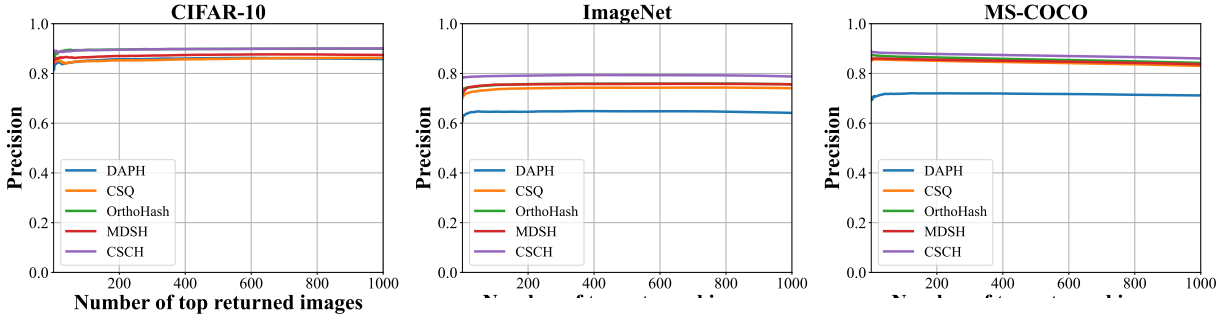


Fig. 5 Precision@top-1K cu

Table 7 Comparisons of mAP with different type consistency loss.

$\mathcal{L}_C$	MS-COCO@5K		
	12 bits	24 bits	32 bits
BCE[44]	0.7591	0.8213	0.8362
Label-smoothing[15]	0.7725	0.8192	0.8370
Ours	**0.7798**	**0.8288**	**0.8440**

Effects of Center-level Consistency. In order t effectiveness of our designed center consistency (14) on the multi-label dataset, we compare it w cross entropy (termed as BCE) loss [44] and cross e with label-smoothing [15] on MS-COCO. Table 7 results when adopting different center consistency losses. Our designed loss achieves the highest mAP at all bits consistently. Within BCE loss, a unique hash centroid is derived for every multi-label image. Cross entropy loss with label-smoothing, on the other hand, computes cosine similarity between hash codes and hash centers as a classification output. By integrating the benefits of these losses, our designed loss yields hash codes with superior quality.

Effects of Hash Centers. To validate the effects of hash centers, we add them to the DAPH method and report the results with 32 bits in Fig. 6. DAPH+$\mathbf{T}_r$ denotes adding randomly assigned hash centers to DAPH. DAPH+$\mathbf{T}_q$ denotes adding optimal hash centers with $\mathbf{A}_q$ assignment to DAPH. We can observe that hash centers significantly improve DAPH's performance, especially on ImageNet, and the optimal hash centers improve the performance even further. This empirically verifies that the central similarity (i.e., hash center) plays an important role in asymmetric retrieval. This global similarity guides query and gallery models to optimize towards the unified objective, and allows the gallery model to get over the restriction of the query model.

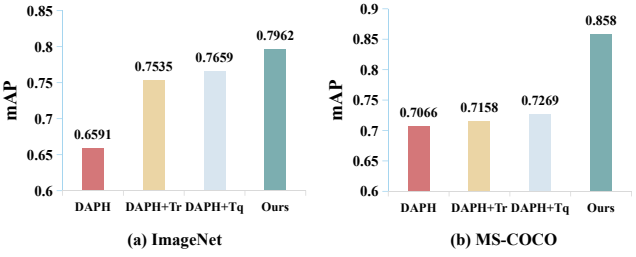
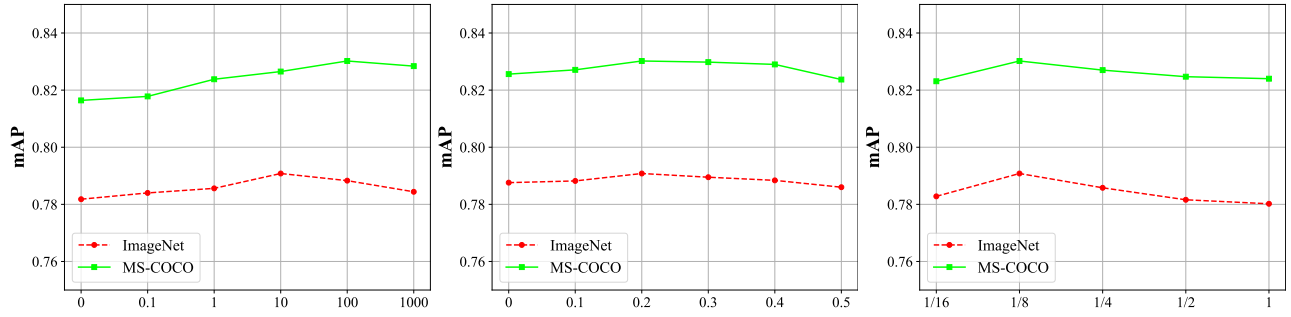


Fig. 6 The mAP comparisons of different methods for 32 bits w/ and w/o hash centers on ImageNet and MS-COCO.

Impact of Image Size. During training phase, we perform random resized crop with crop size of 224×224 for all datasets. To validate the impact of image size on retrieval performance, we adjust the crop size from 160×160 to 256×256. Table 8 shows the mAP comparisons with different backbones on MS-COCO. As the image size increases, there's a noticeable enhancement in retrieval performance. This implies that image dimensions significantly influence the model's performance.

Table 8 The mAP comparisons with different image size.

Query Backone	Gallery Backbone	MS-COCO@5K			
		160×160	196×196	224×224	256×256
MNetv3-S	RN50	0.8134	0.8337	0.8440	0.8494
MNetv3-L	RN101	0.8452	0.8638	0.8705	0.8792



(a) mAP with different γ (b) mAP with different m (c) mAP with different τ

Fig. 7 The mAP changes with different γ, m, τ at 32 bits code length on ImageNet and MS-COCO.

4.6 PR and P@top-K Comparisons

To further assess the retrieval quality of CSCH, we chose to plot the PR curves and precision curves for the top 1K retrieved images at 32 bits. Figures 4 and 5 show that CSCH surpasses all other deep hashing approaches by significant margins in terms of these two evaluation metrics. Importantly, CSCH achieves favorable retrieval results with higher precision at lower recall levels and a larger number of top samples retrieved compared to all other methods. These results highlight the effectiveness of CSCH in real-world retrieval scenarios.

4.7 Parameter Sensitivity

We conduct experiments under different values of γ, m and τ on ImageNet and MS-COCO datasets to further analyze the sensitivity of these parameters. γ is the hyper-parameter that makes a trade-off between center consistency loss and instance consistency loss in Eq. (12). As mentioned in Eq. (9) and Eq. (14), the margin parameter m improves the minimization of intra-class variance between hash codes. And the scale parameter τ controls the size of the hypersphere space associated with feature representation. Figure 7 plots the changes in mAP with different parameter settings at a code length of 32 bits.

As illustrated in Fig. 7 (a), we tune γ in the range of [0, 1000]. We find that the trade-off value of γ is dataset-dependent, and the values of γ differ when obtaining the highest performance on different datasets. In Fig. 7 (b), we record the mAP results by varying m from 0 to 0.5. We can observe that an appropriate m will minimize the intra-class variance and contribute to high-quality hash codes. When $m = 0.2$, our models achieve the best mAP. Finally, we investigate the effects of τ in Eq. (9) and Eq. (14), and we show the mAP results in Fig. 7 (c). τ will affect the density of feature representation within a hypersphere space. Obviously, a relatively smaller τ will bring higher mAP retrieval performance. Empirical results show that $\tau = \frac{1}{8}$ significantly affects our designed center consistency loss, resulting in the highest mAP.

4.8 Visualization

T-SNE Visualization. As illustrated in Fig. 9, we visualize the t-SNE[8] of hash codes generated by large gallery model trained with a large or small one on CIFAR-10. DAPH shows clear degradation in the quality of hash codes, and fails to correctly divide ten groups of images into ten clusters when mutually trained together with a small query model. Instead, CSCH is still able to generate hash codes with good discrimination and visibly distinguishable boundaries, regardless of whether it is trained with a large or small query model. This indicates that our decoupled training approach in CSCH is better suited for asymmetric image retrieval.

Retrieval Results. We randomly select three query images from the single-label dataset ImageNet and the multi-label dataset MS-COCO to conduct the similarity retrieval. Figure 8 illustrates the visualization of the top 5 returned retrieval images of CSCH and DAPH. Our CSCH demonstrates the capability to generate high-quality semantic hash codes, leading to more relevant and desired retrieval results for users.

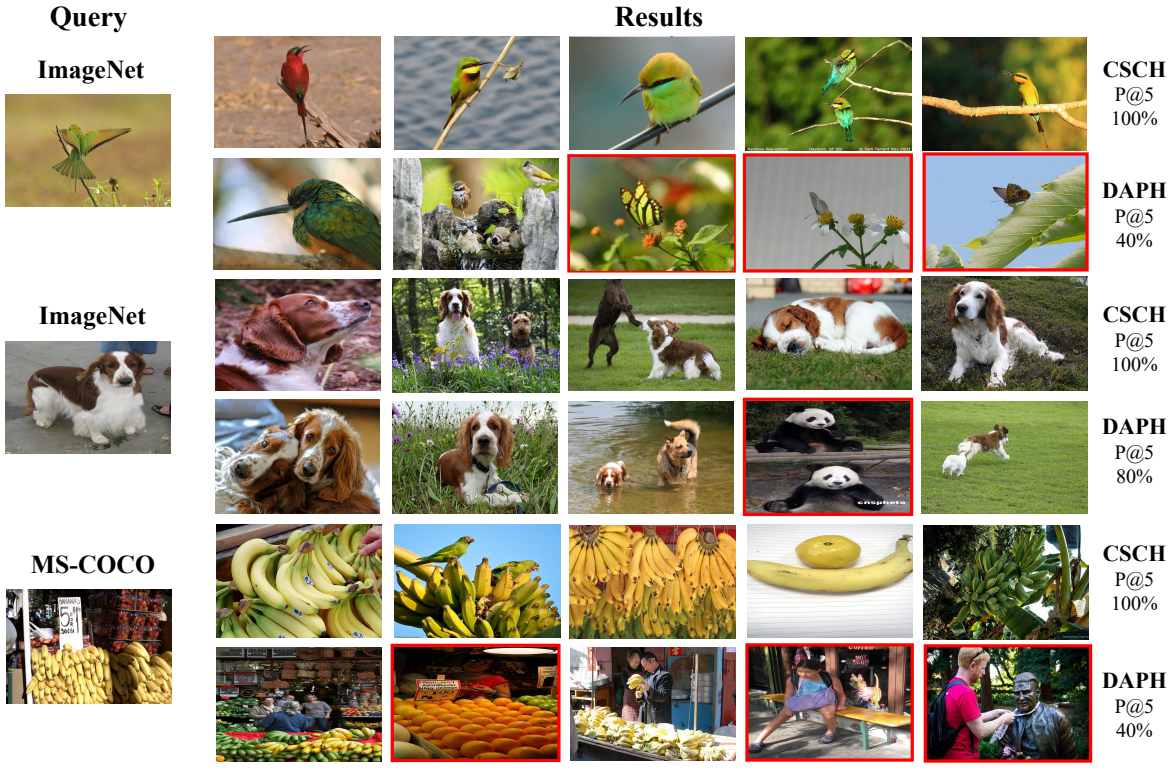
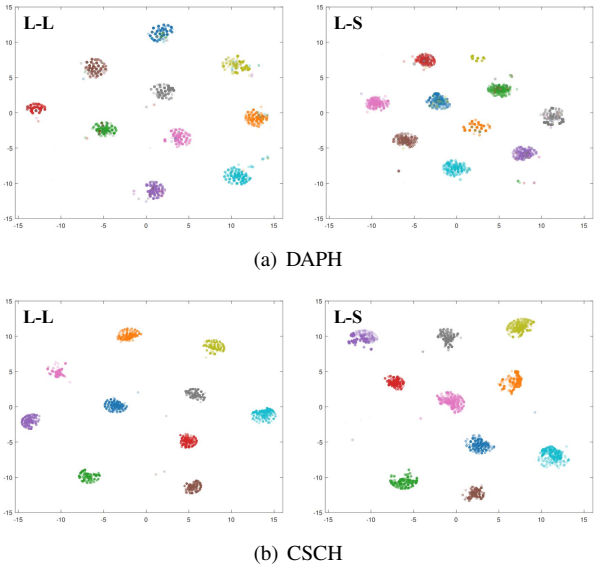


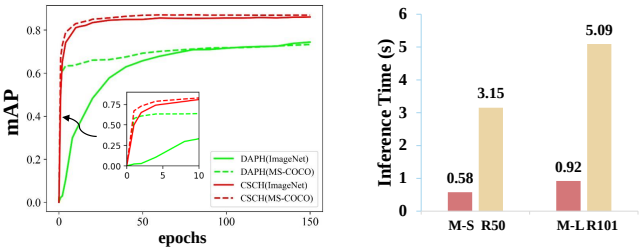
Fig. 8 Top-5 images returned by CSCH and DAPH on ImageNet and MS-COCO. Red boxes denote wrong returned images.



(a) DAPH

(b) CSCH

Fig. 9 The t-SNE of hash codes obtained by the large gallery model is jointly trained with either a large (L-L) or small (L-S) query one.



(a) Training efficiency **(b) Inference efficiency**

Fig. 10 Comparisons of training and inference efficiency.

4.9 Efficiency Comparison

Training Efficiency. We further report the training time of CSCH and DAPH with MNetv3-L and RN101 backbones using $\mathbf{A}_q$ assignment on ImageNet and MS-COCO. The results are plotted in Fig. 10 (a), and the code length is 32. Solid / dotted lines plot the results on ImageNet / MS-COCO. As shown in Fig. 10 (a), our CSCH is able to converge to a better mAP with fewer training epochs under the guidance of optimal hash centers and the large gallery model compared with DAPH.

Inference Efficiency. Under asymmetric image retrieval scenarios, in addition to asymmetric retrieval accuracy, the inference time for queries on edge devices such as mobile phones is a crucial metric. Thus, we employ our CPU to approximate this situation, revealing the inference times for various backbones when processing 100 query images at a 224×224 resolution. As illustrated in Fig. 10 (b), the inference time for MNetv3-S/MNetv3-L is observed to be five times faster as compared to that of RN50/RN101.

5 Conclusion

This paper proposes a novel Central Similarity Consistency Hashing (CSCH) for asymmetric image retrieval. CSCH adopts a small model for the query side with limited computing resources, while utilizing a large model for the gallery side to generate hash codes offline. To the best of our knowledge, CSCH is the first deep hashing approach for asymmetric image retrieval. We first introduce the Hungarian algorithm to optimally align the Hamming similarity of hash centers to the semantic similarity of their classes. Then, we elaborately design the code consistency loss to guarantee the consistency of binary hash codes from both models, while preserving semantic similarity between images. Extensive experiments across three benchmarks consistently demonstrate that the proposed CSCH can significantly enhance the performance of the small query model and surpass state-of-the-art by noticeable margins.

Limitation. Our proposed method CSCH has some limitations. First, the optimal hash centers are generated by a heuristic algorithm, whose Hamming space does not fully align with the semantic space of images. Second, each optimal hash center corresponds to one class label rather than a set of class labels. While we can calculate the hash centroid of a multi-label image, we have not fully taken into account the semantics of the hash centroid. In the future, we will explore how to generate hash centers / centroids with more semantic information for single-label and multi-label images.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant 2022YFB3103500, the National Natural Science Foundation of China under Grants 62106258 and 62202459, and the China Postdoctoral Science Foundation under Grant 2022M713348 and 2022TQ0363, and Young Elite Scientists Sponsorship Program by BAST (NO.BYESS2023304).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Proceedings of the International Conference on Neural Information Processing Systems*, pages 2654–2662, 2014.

[2] Mateusz Budnik and Yannis Avrithis. Asymmetric metric learning for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2021.

[3] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. Deep cauchy hashing for hamming space retrieval. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1229–1237, 2018.

[4] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S. Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5609–5618, 2017.

[5] Xiaohua Chen, Yucan Zhou, Dayan Wu, Chule Yang, Bo Li, Qinghua Hu, and Weiping Wang. Area: Adaptive reweighting via effective area for long-tailed classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19277–19287, 2023.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[8] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31th International Conference on Machine Learning*, pages 647–655, 2014.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[10] Lixin Fan, Kam Woh Ng, Ce Ju, Tianyu Zhang, and Chee Seng Chan. Deep polarized network for supervised learning of accurate binary hashing codes. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 825–831, 2020.

[11] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of International Conference on Very Large Data Bases*, pages 518–529, 1999.

[12] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2916–2929, 2012.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[14] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, 2015.

[15] Jiun Tian Hoe, Kam Woh Ng, Tianyu Zhang, Chee Seng Chan, Yi-Zhe Song, and Tao Xiang. One loss for all: Deep hashing with a single cosine similarity based learning objective. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 24286–24298, 2021.

[16] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc V. Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019.

[17] Young Kyun Jang, Geonmo Gu, ByungSoo Ko, Isaac Kang, and Nam Ik Cho. Deep hash distillation for image retrieval. In *Computer Vision - ECCV 2022 - 17th European Conference*, pages 354–371, 2022.

[18] Qing-Yuan Jiang and Wu-Jun Li. Asymmetric deep supervised hashing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 3342–3349, 2018.

[19] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 1106–1114, 2012.

[22] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2:83–97, 1955.

[23] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3270–3278, 2015.

[24] Tianhong Li, Peng Cao, Yuan Yuan, Lijie Fan, Yuzhe Yang, Rogério Feris, Piotr Indyk, and Dina Katabi. Targeted supervised contrastive learning for long-tailed recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6918, 2022.

[25] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. Feature learning based deep supervised hashing with pairwise labels. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1711–1717, 2016.

[26] Mingbao Lin, Rongrong Ji, Hong Liu, and Yongjian Wu. Supervised online hashing via hadamard codebook learning. In *Proceedings of the 2018 ACM Multimedia Conference on Multimedia Conference*, pages 1635–1643, 2018.

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference*, volume 8693, pages 740–755, 2014.

[28] Venice Erin Liong, Jiwen Lu, Yap-Peng Tan, and Jie Zhou. Deep video hashing. *IEEE Trans. Multim.*, 19(6):1209–1219, 2017.

[29] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2064–2072, 2016.

[30] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[31] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the International Conference on Neural Information Processing Systems*, pages 8024–8035, 2019.

[33] Fumin Shen, Xin Gao, Li Liu, Yang Yang, and Heng Tao Shen. Deep asymmetric pairwise hashing. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1522–1530, 2017.

[34] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[35] Liangdao Wang, Yan Pan, Cong Liu, Hanjiang Lai, Jian Yin, and Ye Liu. Deep hashing with minimal-distance-separated hash centers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23455–23464, 2023.

[36] Lin Wang, Wanqian Zhang, Dayan Wu, Fei Zhu, and Bo Li. Attack is the best defense: Towards preemptive-protection person re-identification. In *Proceedings of the 30st ACM International Conference on Multimedia*, pages 550–559, 2022.

[37] Yair Weiss, Antonio Torralba, and Robert Fergus. Spectral hashing. In *Proceedings of the International Conference on*

Neural Information Processing Systems, pages 1753–1760, 2008.

[38] Dayan Wu, Qi Dai, Jing Liu, Bo Li, and Weiping Wang. Deep incremental hashing network for efficient image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9069–9077, 2019.

[39] Dayan Wu, Qinghang Su, Bo Li, and Weiping Wang. Efficient hash code expansion by recycling old bits. In *Proceedings of the 30st ACM International Conference on Multimedia*, pages 572–580, 2022.

[40] Hui Wu, Min Wang, Wengang Zhou, Houqiang Li, and Qi Tian. Contextual similarity distillation for asymmetric image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9479–9488, 2022.

[41] Hui Wu, Min Wang, Wengang Zhou, Zhenbo Lu, and Houqiang Li. Asymmetric feature fusion for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11082–11092, 2023.

[42] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2156–2162. AAAI Press, 2014.

[43] Zexian Yang, Dayan Wu, Wanqian Zhang, Bo Li, and Weiping Wang. Handling label uncertainty for camera incremental person re-identification. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 6253–6263, 2023.

[44] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis E. H. Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3080–3089, 2020.

[45] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Trans. Image Process.*, 24:4766–4779, 2015.

[46] Wanqian Zhang, Dayan Wu, Yu Zhou, Bo Li, Weiping Wang, and Dan Meng. Binary neural network hashing for image retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 317–326, 2021.

[47] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[48] Cairong Zhao, Yuanpeng Tu, Zhihui Lai, Fumin Shen, Heng Tao Shen, and Duoqian Miao. Salience-guided iterative asymmetric mutual hashing for fast person re-identification. *IEEE Trans. Image Process.*, 30:7776–7789, 2021.

[49] Shu Zhao, Dayan Wu, Wanqian Zhang, Yu Zhou, Bo Li, and Weiping Wang. Asymmetric deep hashing for efficient hash code compression. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 763–771, 2020.

[50] Shu Zhao, Dayan Wu, Yucan Zhou, Bo Li, and Weiping Wang. Rescuing deep hashing from dead bits problem. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1338–1344, 2021.

[51] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2415–2421, 2016.

Zhaofeng Xuan received his B.S. degree from Shandong University in 2021. He is currently pursuing a master degree in the Institute of Information Engineering, and the School of Cyber Security, University of Chinese Academy of Sciences. His research interests include multimedia retrieval and computer vision.

Dayan Wu received the B.S. degree from the Huazhong University of Science and Technology in 2014 and the Ph.D. degree from the University of Chinese Academy of Sciences in 2019. From 2018 to 2019, he worked as a Research Intern with Microsoft Research Asia. He is currently an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His research interests include multimedia retrieval and indexing, visual tracking, and computer vision.

Wanqian Zhang received the B.S. degree from the University of Science and Technology Beijing in 2011 and the Ph.D. degree from the Institute of Information Engineering, Chinese Academy of Sciences, in 2021. He is currently a Tenure-Track Assistant Professor with the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include computer vision, multimedia retrieval, and deep hashing their applications in industry.

Qinghang Su received his B.S. degree from Shandong University in 2020. He is currently pursuing a Ph.D. degree in the Institute of Information Engineering, and the School of Cyber Security, University of Chinese Academy of Sciences. His research interests include multimedia retrieval and computer vision.

Bo Li received the Ph.D. degree from the University of Chinese Academy of Sciences in 2011. He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His research interests include big data storage, multimedia retrieval, and data security.

Weiping Wang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China. He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. He has more than 100 publications in major journals and international conferences. His research interests include big data, data security, and artificial intelligence.