Applied Intelligence
https://doi.org/10.1007/s10489-022-04189-6

Online supervised collective matrix factorization hashing for cross-modal retrieval

Zhenqiu Shu[1] · Li Li[1] · Jun Yu[2] · Donglin Zhang[3] · Zhengtao Yu[1] · Xiao-Jun Wu[3]

Accepted: 17 September 2022

Abstract
Recently, online hashing has received extensive attention in cross-modal retrieval since it can effectively deal with large-scale streaming data. However, some of them in online cross-modal retrieval still have the following limitations: (1) They cannot take full advantage of the supervision information among the data in some cases. (2) The hash codes learned from online hashing methods only preserve the shared properties of multi-modal data, and neglect the specific properties of each modality. To solve the above issues, we propose a novel supervised online cross-modal retrieval hashing method, i.e., Online Supervised Collective Matrix Factorization Hashing(OSCMFH). OSCMFH utilizes semantic labels to obtain the shared and specific latent semantic representation of all modalities of new data. Meanwhile, the latent semantic representation of old data is adaptively updated by the newly updated hash model without accessing the old data. Furthermore, the hash codes can be generated by quantifying the combination of the latent semantic representation of new and old data. Therefore, the hash codes not only embed the semantic information of multi-modality data but also keep the shared and specific properties of multi-modal data. We conduct extensive experiments on four benchmark datasets to demonstrate the effectiveness of our OSCMFH algorithm compared with other state-of-the-art online learning algorithms.

Keywords Cross-modal retrieval · Online hashing · Streaming data · Collective matrix factorization · Supervised learning · Hash codes

1 Introduction

In the field of big data, multimedia data is growing explosively in the past few years. The cross-modal hashing (CMH) methods have attracted much attention in large-scale data retrieval due to their efficiency and low storage space. Since the binary hash code is a short number of bits, it can directly perform the XOR operation to calculate the similarity between data. Therefore, CMH methods measure the similarity between modalities by encoding different modality data into the compact binary hash codes and then exploiting their distances in Hamming space. The methods greatly reduce the storage requirements of data and improve the cross-modal retrieval efficiency. At present, many existing CMH methods have achieved excellent performance in retrieval tasks [1–6]. However, most of them adopt the batch-based way to learn the hash codes for all multi-modal data. This way exists some drawbacks when dealing with large-scale streaming data. For example, for ever-increasing streaming multi-modal data, the batch-based hashing approaches need to retrain all data points to learn the hash codes of the new data points. Therefore, when new streaming data arrive frequently, it brings an unacceptable amount of computational cost and memory costs.

In recent years, online hashing retrieval methods show higher efficiency in massive streaming multi-modal data retrieval than offline hashing methods. Existing online CMH methods can be simply divided into unsupervised and supervised methods according to whether the semantic labels are utilized. Online Cross-Modal Hashing (OCMH)

✉ Zhenqiu Shu
shuzhenqiu@163.com

[1] School of Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China

[2] The College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China

[3] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi, China

Published online: 22 October 2022

[7] decomposes the feature matrix into a shared latent semantic matrix and a transition matrix that supports online learning by the matrix factorization method. Then hash codes are learned through the shared latent semantic matrix. Online Collective Matrix Factorization Hashing (OCMFH) [8] learns the latent semantic representation of each modality in the training data through collaborative matrix factorization, and it learns hash functions and hash codes in an online learning manner. On the contrary, online supervised cross-modal hashing methods use label information to guide the learning of hash functions. Therefore, the discriminative ability of the hash codes is enhanced, and the performance is more effective than unsupervised methods. Online Latent Semantic Hashing (OLSH) [9] maps the learned hash codes and discrete labels of data to a continuous latent semantic space, so that relative semantics can more accurately measure the distance of data points. Online Label Consistent Hashing(OLCH) [10] progressively learns the hash codes of the current newly arrived data, and updates the hash functions in a streaming manner. Flexible Online Multi-modal Hashing (FOMH) [11] flexibly projects multi-modal data into binary hash codes via online supervised hashing. Furthermore, it adaptively learns the weight of each modality to capture the changes in streaming data in time.

In this work, we propose an online hashing approach, called Online Supervised Collective Matrix Factorization Hashing (OSCMFH), for cross-modal retrieval. Figure 1 shows the framework of our OSCMFH approach. Specifically, the proposed OSCMFH method directly guides hash learning through the semantic labels to preserve more semantic information. Moreover, the share and specific properties of the multimodal data are obtained by using the matrix factorization method. In addition, an online optimization strategy is proposed to realize online adaptive updating of latent semantic representations of both old and new data. Finally, we can generate the hash codes by quantifying the combination of the latent semantic representations of both old and new data. The cross-modal retrieval results on several datasets have demonstrated the effectiveness of our OSCMFH approach.

The main contributions of this paper can be summarized as follows:

1. Our OSCMFH approach is a supervised learning method, which employs the supervised information to decompose the multi-media data into the shared and specific latent semantic representation. Therefore, the obtained hash codes keep the common and specific properties of multi-media data.

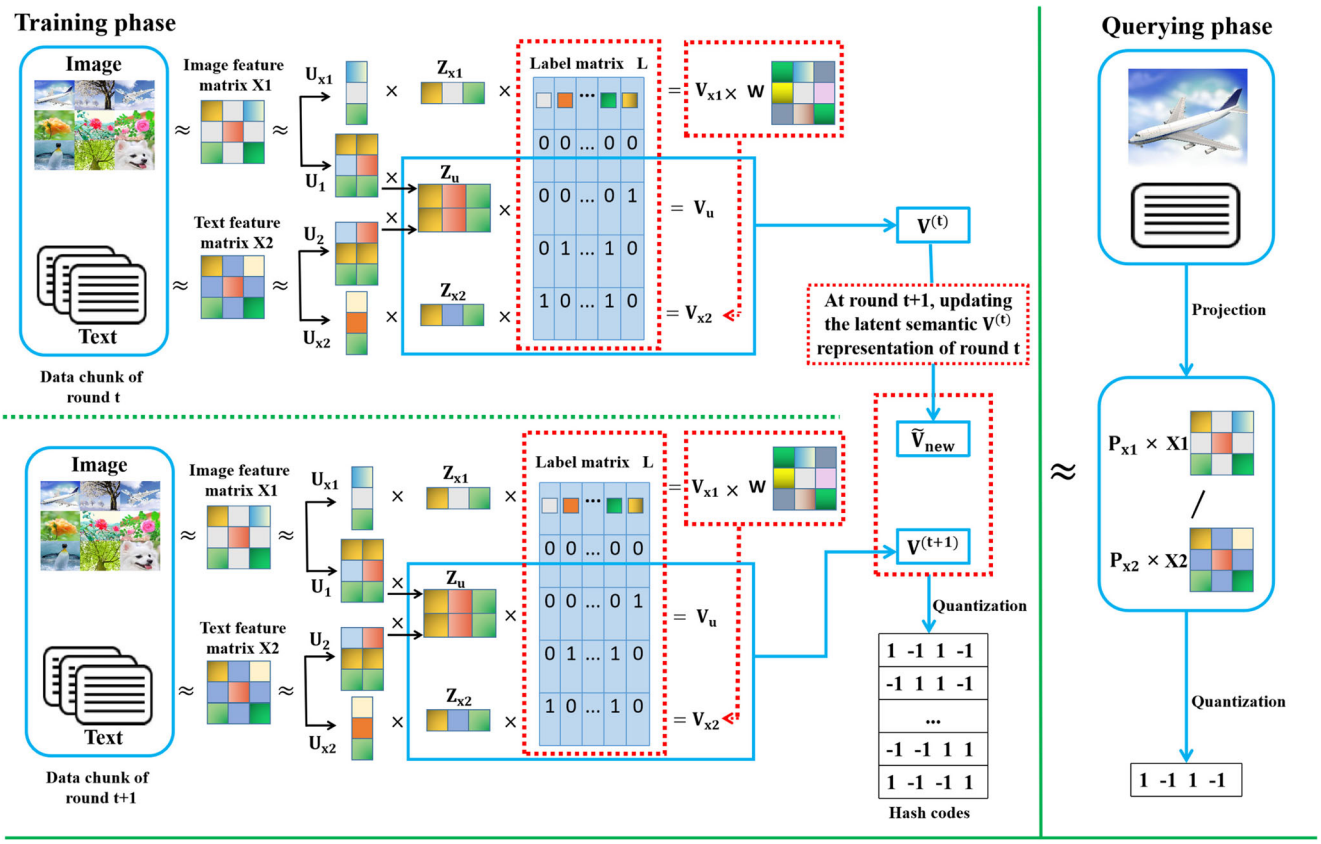


Fig. 1 The framework of our proposed OSCMFH approach

2. We adopt an online learning manner to update the hash codes of old data. Therefore, it can fit the new hashing model without accessing old data. Thus, the proposed method can take less computational cost and lower storage space than traditional offline hashing methods.
3. An efficient online hashing optimization method is developed to solve the proposed model. Extensive experimental results on four benchmark datasets verify the superiority of our OSCMFH method.

The remainder of the paper is arranged as follows: Section 2 introduces the related works of cross-modal retrieval. Section 3 details our approach. Section 4 presents the experimental result and the analysis. Section 5 concludes this work.

2 Related works

2.1 Offline hashing

At present, most of the cross-modal hashing retrieval methods adopt the offline learning manner. Many studies have shown that they can achieve excellent performances in large-scale cross-modal data retrieval. Wang et al. [12] proposed to obtain the shared latent semantic space of all modalities and the modality-specific latent semantic space through matrix factorization. Thus, the learned hash codes can preserve the shared and specific properties of multimodal data. Cluster-wise unsupervised hashing [13] proposed a clustering scheme to jointly learn compact hash codes and hash functions. The semantic consistency between modalities and the similarity within modalities are guaranteed. Discrete Matrix Factorization Hashing (DMFH) [14] performs matrix decomposition on the similarity graph of each modality, and directly extracts the discrete common hashing representation features of each modality. Therefore, the quantitative loss caused by hashing relaxation is significantly reduced. Yao et al. [15] explore the shared and modality-specific properties of multi-modal data through matrix factorization. The aforementioned methods are unsupervised learning methods and neglect the supervision information among the multimedia datasets. In general, supervised cross-modal hashing methods have achieved better retrieval accuracy than unsupervised methods since they take full advantage of the semantic labels or similarities of the multi-modality data.

Various supervised hashing learning approaches have developed in recent years. Li et al.[16] proposed leveraging semantic label correlations to guide latent feature learning. Simultaneously, the hash codes are generated in a discrete manner. Furthermore, pair-wise semantic similarity is preserved by constructing a similarity matrix whose size is the square of the training data. Since a squared similarity matrix is constructed, it consumes a lot of time and space. Discrete Asymmetric Hashing(DAH) [17] can handle equal or unequal lengths of hash codes in retrieval tasks, which ensures the flexibility of retrieval. Multi-hash codes joint learning (MOON) [18] simultaneously learns hash codes of different lengths in the cross-modal retrieval process, thus reducing the amount of computation. Wang et al. [19] proposed to transform the multi-modality data into a shared latent semantic representation, and the hash learning is guided by the semantic labels. A Matrix Tri-Factorization Hashing (MTFH) [20] flexibly learns the modal-specific hash codes with the same or different lengths through a semantic correlation matrix, which can show great flexibility in various retrieval tasks. However, the offline hashing methods need unacceptable time and storage space costs in retrieval tasks for continuously arriving streaming data in most cases. Therefore, the offline hashing methods are unsuitable for dealing with streaming data.

2.2 Online hashing

Online hashing approaches are very efficient in processing large-scale streaming data. The existing online hashing methods can be roughly divided into online single-modal hashing methods and online cross-modal hashing methods according to data modalities.

Online single-modal hashing methods [21–30] were proposed for single-modality retrieval. It mainly solves the problem of streaming single-modal data retrieval. Online Sketching Hashing (OSH) [21] extracts summary information from the acquired new data. In addition, the hash function is learned from the summary information matrix, which can solve the problem that hash model training requires a large amount of computation in a large-scale dataset. Online Kernel-based Hashing (OKH) [24] determines whether the current hash model needs to be updated by comparing the Hamming distance of the hash codes with a given threshold. Lin et al. [28] proposed to update the hash functions in a class-level manner to achieve fast online adaptive updates. Hadamard Matrix Guided Online Hashing (HMOH) [29] employs the Sylvester method to construct Hadamard matrices of arbitrary order. Furthermore, each column of the Hadamard matrix is used as the target code for each class label to guide the hashing learning.

Online cross-modal hashing methods [7–11, 31–33] retrieve semantically related instances of another modality by taking one modality as input to a query (e.g., using an image as query data to retrieve semantically related text instances). Online Cross-Modal Hashing (OCMH) [7] employs the collaborative matrix factorization algorithm to learn the latent semantic representation of each modality

in the training data, and the hash functions and the hash codes are obtained by the online learning manner. OCMH is an unsupervised online learning method, which may affect retrieval accuracy due to insufficient semantic association between modalities. Flexible Online Multi-modal Hashing (FOMH) [11] flexibly projects multi-modal data into binary hash codes via online supervised hashing. Moreover, the weights of each modality are adaptively learned to capture changes in the streaming data in time. Discrete Online Cross-modal Hashing (DOCH) [32] directly exploits the similarity between new and old data in Hamming space, and embeds the semantic information into hashing learning. However, DOCH adopts a bit-by-bit discrete online updating method to learn hash codes through multiple iterations, and thus consumes expensive time costs in the optimization procedure. Lu et al. [33] proposed a parameter-free online hashing model to accurately capture the variation of different queries through query-adaptive and self-weighted. Furthermore, the discrimination ability of pairwise semantic hash codes can be improved by associating the learned hash codes with similar semantic information. However, both FOMH and OMHDQ cannot adaptively update the hash codes of the old data after learning the hash codes of the newly arrived data stream. Therefore, their retrieval performances may be affected in real tasks.

3 Our proposed method

In this section, we describe our proposed OSCMFH method in detail. To simplify the description, we only describe multi-modal data composed of images and text.

3.1 Problem definition

At each round t, a new image-text pairwise data chunk $X^{(t)} = [X_1^{(t)}, X_2^{(t)}]$ is added into the training dataset. Here, $X_1^{(t)} = \left[x_1^{1(t)}, x_1^{2(t)}, \ldots, x_1^{N_t(t)}\right] \in \mathbb{R}^{d_1 \times N_t}$ and $X_2^{(t)} = \left[x_2^{1(t)}, x_2^{2(t)}, \ldots, x_2^{N_t(t)}\right] \in \mathbb{R}^{d_2 \times N_t}$ respectively represents the feature matrices of the image modality and the text modality, where d_1 and d_2 are the dimensionlity of the image and text modality, respectively, and N_t is the size of the new image-text pairwise data chunk. Let $L^{(t)} \in \{0,1\}^{c \times N_t}$ denote the label matrix, where c is the number of classes. Before round t, the old data chunk $\widetilde{X}^{(t-1)} = [\widetilde{X}_1^{(t-1)}, \widetilde{X}_2^{(t-1)}]$ contains $\widetilde{N}_{t-1}$ training samples and its label matrix $\widetilde{L}^{(t-1)} \in \{0,1\}^{c \times \widetilde{N}_{t-1}}$. Therefore, the total training dataset in round t is denoted as $X = [\widetilde{X}^{(t-1)}, X^{(t)}]$ containing $N = N_t + \widetilde{N}_{t-1}$ data points. Its total label matrix is denoted as $L = \left[\widetilde{L}^{(t-1)}, L^{(t)}\right] \in \{0,1\}^{c \times N}$ and total hash codes are denoted as $B \in \{-1,1\}^{k \times N}$, where k is the hash codes length. Given a query data X_{1quey} or X_{2quey}, their hash codes are available via $f_1 = sgn(P_{x1}X_{1quey})$ and $f_2 = sgn(P_{x2}X_{2quey})$, where $P_{x1} \in \mathbb{R}^{k \times d_1}$ and $P_{x2} \in \mathbb{R}^{k \times d_2}$ are the projection matrices of the image modality and the text modality, respectively. $sgn(\cdot)$ is a symbolic function.

3.2 Model formulation

In cross-modal retrieval, multi-modal data include both the common features and their special features. Most of the cross-modal retrieval methods describe the multi-modal data by the share latent semantic representations, and completely neglect the special part of each modality. Inspired by Joint and Individual Matrix Factorization Hashing (JIMFH) [12], our method obtains a shared latent semantic representation $V_u \in \mathbb{R}^{k_1 \times N}$ of the multi-modality data and the modality-specific latent semantic representation $V_{x1} \in \mathbb{R}^{k_2 \times N}$ and $V_{x2} \in \mathbb{R}^{k_2 \times N}$ of the image modality and the text modality via matrix factorization. Moreover, the label constraints are imposed on the representation matrices V_u, V_{x1} and V_{x2} to further improve the retrieval accuracy. We can accomplish by introducing auxiliary matrices $Z_u \in \mathbb{R}^{k_1 \times c}$, $Z_{x1} \in \mathbb{R}^{k_2 \times c}$ and $Z_{x2} \in \mathbb{R}^{k_2 \times c}$, i.e., $V_u = Z_u L$, $V_{x1} = Z_{x1} L$ and $V_{x2} = Z_{x2} L$. Therefore, our objective function is given as follows:

$$\begin{aligned} \mathcal{L} = & \sum_{i=1}^{2} \lambda_i \left\| X_i - U_i Z_u L \right\|_F^2 + \sum_{i=1}^{2} \lambda_i \left\| X_i - U_{xi} Z_{xi} L \right\|_F^2 \\ & + \mu \left\| Z_{x2} L - W Z_{x1} L \right\|_F^2 + \gamma R \\ & s.t. \sum_{i=1}^{2} \lambda_i = 1, \end{aligned} \tag{1}$$

where $R = \sum_{i=1}^{2} \left(\|U_i\|_F^2 + \|U_{xi}\|_F^2 + \|Z_{xi} L\|_F^2 \right) + \|Z_u L\|_F^2 + \|W\|_F^2$ is a regularization term to avoid overfitting. $U_1 \in \mathbb{R}^{d_1 \times k_1}$ and $U_2 \in \mathbb{R}^{d_2 \times k_1}$ denote the image and text latent semantic basis matrices, respectively. $U_{x1} \in \mathbb{R}^{d_1 \times k_2}$ and $U_{x2} \in \mathbb{R}^{d_2 \times k_2}$ are the basis matrices of the image and text, respectively. k_1 and k_2 are the dimensionalities of the share latent semantic representation matrix and the specific latent semantic matrix, respectively. Furthermore, k_1 and k_2 are positive integers less than k, and $k = k_1 + k_2$. $\lambda_i \in (0,1)$ is a weight parameter that balances the image and text modalities. $W \in \mathbb{R}^{k_2 \times k_2}$ is the relationship matrix between two modalities. μ and γ denote the trade-off parameters to control the corresponding contribution term. By imposing the constraint term $\|Z_{x2} L - W Z_{x1} L\|_F^2$, the specific representations of the image and text modalities are related. Therefore, the latent semantic representation $V \in$

$\mathbb{R}^{k\times N}$ of the training data can be represented as follows:

$$V = \begin{bmatrix} Z_u L \\ Z_{x2} L \end{bmatrix}. \tag{2}$$

Our proposed OSCMFH method is designed to deal with the streaming data through an online learning manner. While hash codes are generated for the newly arrived data, the hash codes for old data are also updated adaptively. Especially, in round t, when a new data chunk arrives, the online learning procedure of our OSCMFH approach can be described as follows: 1) Updating the hash functions of each modality and generating a new latent semantic representation with label information; 2) Updating the latent semantic representation of old data chunk of $(t-1)$-th round; 3) Generating the hash codes of the new and old data by quantifying the combination of the latent semantic representation. Therefore, the total training data for online learning includes new arrival data and old accumulated data as follows:

$$\begin{aligned} \mathcal{L}_{new} = & \sum_{i=1}^{2} \lambda_i \left\| X_i^{(t)} - U_i^{(t)} Z_u^{(t)} L^{(t)} \right\|_F^2 + \\ & \sum_{i=1}^{2} \lambda_i \left\| X_i^{(t)} - U_{xi}^{(t)} Z_{xi}^{(t)} L^{(t)} \right\|_F^2 + \\ & \mu \left\| Z_{x2}^{(t)} L^{(t)} - W^{(t)} Z_{x1}^{(t)} L^{(t)} \right\|_F^2 + \gamma R_1 \\ & s.t. \sum_{i=1}^{2} \lambda_i = 1, \end{aligned} \tag{3}$$

where $R_1 = \sum_{i=1}^{2} \left(\left\| U_i^{(t)} \right\|_F^2 + \left\| U_{xi}^{(t)} \right\|_F^2 + \left\| Z_{xi}^{(t)} L^{(t)} \right\|_F^2 \right) + \left\| Z_u^{(t)} L^{(t)} \right\|_F^2 + \left\| W^{(t)} \right\|_F^2$.

$$\begin{aligned} \widetilde{\mathcal{L}}_{old} = & \sum_{i=1}^{2} \lambda_i \left\| \widetilde{X}_i^{(t-1)} - U_i^{(t)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \\ & \sum_{i=1}^{2} \lambda_i \left\| \widetilde{X}_i^{(t-1)} - U_{xi}^{(t)} \widetilde{Z}_{xi}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \\ & \mu \left\| \widetilde{Z}_{x2}^{(t-1)} \widetilde{L}^{(t-1)} - W^{(t)} \widetilde{Z}_{x1}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \gamma R_2 \\ & s.t. \sum_{i=1}^{2} \lambda_i = 1, \end{aligned} \tag{4}$$

where $R_2 = \sum_{i=1}^{2} \left\| \widetilde{Z}_{xi}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \left\| \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2$.

Therefore, the overall objective function of our proposed OSCMFH approach is represented as follows:

$$\mathcal{L} = \mathcal{L}_{new} + \widetilde{\mathcal{L}}_{old}. \tag{5}$$

The objective function (5) is non-convex for all variables. However, it is convex for any one variable by fixing other variables in the model.

3.3 Online optimization

In this subsection, an iterative updating scheme is used to solve (5). Specially, we can optimize one variable by fixing other variables. The detailed optimization process is given as follows:

1) Update $U_1^{(t)}$ by fixing other variables, (5) can be rewritten as follows:

$$\begin{aligned} \mathcal{L} = & \lambda \left\| X_1^{(t)} - U_1^{(t)} Z_u^{(t)} L^{(t)} \right\|_F^2 + \\ & \lambda \left\| \widetilde{X}_1^{(t-1)} - U_1^{(t)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \gamma \left\| U_1^{(t)} \right\|_F^2. \end{aligned} \tag{6}$$

By setting $\frac{\partial \mathcal{L}}{\partial U_1^{(t)}} = 0$, we can obtain the following updating rules:

$$U_1^{(t)} = E_1^{(t)} \left(C_1^{(t)} + \frac{\gamma}{\lambda} I \right)^{-1}, \tag{7}$$

where $E_1^{(t)}$ can be given as follows:

$$\begin{aligned} E_1^{(t)} &= X_1^{(t)} \left(Z_u^{(t)} L^{(t)} \right)^T + \widetilde{X}_1^{(t-1)} \left(\widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right)^T \\ &= X_1^{(t)} \left(Z_u^{(t)} L^{(t)} \right)^T + \widetilde{E}_1^{(t-1)}, \end{aligned} \tag{8}$$

and $C_1^{(t)}$ is given as follows:

$$\begin{aligned} C_1^{(t)} &= Z_u^{(t)} L^{(t)} \left(Z_u^{(t)} L^{(t)} \right)^T \\ &\quad + \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \left(\widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right)^T \\ &= Z_u^{(t)} L^{(t)} \left(Z_u^{(t)} L^{(t)} \right)^T + \widetilde{C}_1^{(t-1)}. \end{aligned} \tag{9}$$

It can be seen from (8) and (9) that both $E_1^{(t)}$ and $C_1^{(t)}$ consist of two parts. The former part is related to newly arrived data in the t-th round, and the latter part is related to the old data accumulated before the t-th round. Therefore, $E_1^{(t)}$ and $C_1^{(t)}$ can be calculated incrementally, and $U_1^{(t)}$ can be updated through online learning. Similarly, $U_2^{(t)}$, $U_{x1}^{(t)}$, $U_{x2}^{(t)}$ and $W^{(t)}$ can also be updated through online learning.

2) Update $U_2^{(t)}$ by fixing other variables. Similar to $U_1^{(t)}$, we can obtain the following updating rule:

$$U_2^{(t)} = E_2^{(t)} \left(C_2^{(t)} + \frac{\gamma}{1-\lambda} I \right)^{-1}, \tag{10}$$

$$E_2^{(t)} = X_2^{(t)}\left(Z_u^{(t)}L^{(t)}\right)^T + \widetilde{X}_2^{(t-1)}\left(\widetilde{Z}_u^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = X_2^{(t)}\left(Z_u^{(t)}L^{(t)}\right)^T + \widetilde{E}_2^{(t-1)}, \quad (11)$$

$$C_2^{(t)} = C_1^{(t)}. \quad (12)$$

3) Update $U_{x1}^{(t)}$ by fixing other variables. Thus, (5) can be rewritten as follows:

$$\mathcal{L} = \lambda\left\|X_1^{(t)} - U_{x1}^{(t)}Z_{x1}^{(t)}L^{(t)}\right\|_F^2 + \lambda\left\|\widetilde{X}_1^{(t-1)} - U_{x1}^{(t)}\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right\|_F^2 + \gamma\left\|U_{x1}^{(t)}\right\|_F^2. \quad (13)$$

Let $\frac{\partial\mathcal{L}}{\partial U_{x1}^{(t)}} = 0$, we have

$$U_{x1}^{(t)} = E_3^{(t)}\left(C_3^{(t)} + \frac{\gamma}{\lambda}I\right)^{-1}, \quad (14)$$

$$E_3^{(t)} = X_1^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{X}_1^{(t-1)}\left(\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = X_1^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{E}_3^{(t-1)}, \quad (15)$$

$$C_3^{(t)} = Z_{x1}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\left(\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = Z_{x1}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{C}_3^{(t-1)}. \quad (16)$$

4) Update $U_{x2}^{(t)}$ by fixing other variables. Similar to $U_{x1}^{(t)}$, we have

$$U_{x2}^{(t)} = E_4^{(t)}\left(C_4^{(t)} + \frac{\gamma}{1-\lambda}I\right)^{-1}, \quad (17)$$

$$E_4^{(t)} = X_2^{(t)}\left(Z_{x2}^{(t)}L^{(t)}\right)^T + \widetilde{X}_2^{(t-1)}\left(\widetilde{Z}_{x2}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = X_2^{(t)}\left(Z_{x2}^{(t)}L^{(t)}\right)^T + \widetilde{E}_4^{(t-1)}, \quad (18)$$

$$C_4^{(t)} = Z_{x2}^{(t)}L^{(t)}\left(Z_{x2}^{(t)}L^{(t)}\right)^T + \widetilde{Z}_{x2}^{(t-1)}\widetilde{L}^{(t-1)}\left(\widetilde{Z}_{x2}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = Z_{x2}^{(t)}L^{(t)}\left(Z_{x2}^{(t)}L^{(t)}\right)^T + \widetilde{C}_4^{(t-1)}. \quad (19)$$

5) Update $W^{(t)}$ by fixing other variables. Thus, (5) can be rewritten as follows:

$$\mathcal{L} = \mu\left\|Z_{x2}^{(t)}L^{(t)} - W^{(t)}Z_{x1}^{(t)}L^{(t)}\right\|_F^2 + \gamma\left\|W^{(t)}\right\|_F^2 + \mu\left\|\widetilde{Z}_{x2}^{(t-1)}\widetilde{L}^{(t-1)} - W^{(t)}\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right\|_F^2. \quad (20)$$

Let $\frac{\partial\mathcal{L}}{\partial W^{(t)}} = 0$, we have

$$W^{(t)} = F^{(t)}\left(D^{(t)} + \frac{\gamma}{\mu}I\right)^{-1}, \quad (21)$$

$$F^{(t)} = Z_{x2}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{Z}_{x2}^{(t-1)}\widetilde{L}^{(t-1)}\left(\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = Z_{x2}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{F}^{(t-1)}, \quad (22)$$

$$D^{(t)} = Z_{x1}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\left(\widetilde{Z}_{x1}^{(t-1)}\widetilde{L}^{(t-1)}\right)^T = Z_{x1}^{(t)}L^{(t)}\left(Z_{x1}^{(t)}L^{(t)}\right)^T + \widetilde{D}^{(t-1)}. \quad (23)$$

6) Update $Z_u^{(t)}$ by fixing other variables. Therefore, we can rewrite (5) as follow:

$$\mathcal{L} = \lambda\left\|X_1^{(t)} - U_1^{(t)}Z_u^{(t)}L^{(t)}\right\|_F^2 + \gamma\left\|Z_u^{(t)}L^{(t)}\right\|_F^2 + (1-\lambda)\left\|X_2^{(t)} - U_2^{(t)}Z_u^{(t)}L^{(t)}\right\|_F^2. \quad (24)$$

Let $\frac{\partial\mathcal{L}}{\partial Z_u^{(t)}} = 0$, we have

$$Z_u^{(t)} = \left(\lambda U_1^{(t)T}U_1^{(t)} + (1-\lambda)U_2^{(t)T}U_2^{(t)} + \gamma I\right)^{-1}\left(\lambda U_1^{(t)T}X_1^{(t)}L^{(t)T} + (1-\lambda)U_2^{(t)T}X_2^{(t)}L^{(t)T}\right)\left(L^{(t)}L^{(t)T}\right)^{-1}. \quad (25)$$

7) Update $Z_{x1}^{(t)}$ by fixing other variables. Equation (5) can be rewritten as follows:

$$\mathcal{L} = \lambda\left\|X_1^{(t)} - U_{x1}^{(t)}Z_{x1}^{(t)}L^{(t)}\right\|_F^2 + \gamma\left\|Z_{x1}^{(t)}L^{(t)}\right\|_F^2 + \mu\left\|Z_{x2}^{(t)}L^{(t)} - W^{(t)}Z_{x1}^{(t)}L^{(t)}\right\|_F^2. \quad (26)$$

Let $\frac{\partial\mathcal{L}}{\partial Z_{x1}^{(t)}} = 0$, we have

$$Z_{x1}^{(t)} = \left(\lambda U_{x1}^{(t)T}U_{x1}^{(t)} + \mu W^{(t)T}W^{(t)} + \gamma I\right)^{-1}\left(\lambda U_{x1}^{(t)T}X_1^{(t)}L^{(t)T} + \mu W^{(t)T}Z_{x2}^{(t)}L^{(t)}L^{(t)T}\right)\left(L^{(t)}L^{(t)T}\right)^{-1}. \quad (27)$$

8) Update $Z_{x2}^{(t)}$ by fixing other variables. Therefore, the updating rule of $Z_{x1}^{(t)}$ is given as follows:

$$Z_{x2}^{(t)} = \left((1-\lambda)\, U_{x2}^{(t)T} U_{x2}^{(t)} + (\mu+\gamma)\, I\right)^{-1} \left((1-\lambda)\, U_{x2}^{(t)T} X_2^{(t)} L^{(t)T} + \mu W^{(t)} Z_{x1}^{(t)} L^{(t)} L^{(t)T}\right)\left(L^{(t)} L^{(t)T}\right)^{-1}. \quad (28)$$

The final solution can be obtained by iteratively updating the rules of various variables until the objective function converges or reaches the maximum number of iterations. According to (2), we can get the latent semantic representation $V^{(t)}$ of t-th round as follows:

$$V^{(t)} = \begin{bmatrix} Z_u^{(t)} L^{(t)} \\ Z_{x2}^{(t)} L^{(t)} \end{bmatrix}. \quad (29)$$

It can be seen that the matrix $V^{(t)}$ can be updated after the arrival of new data in the t-th round. Meanwhile, the hash model in the $(t-1)$-th round is updated without accessing the old data while the t-th round data arrives. Furthermore, hash codes of old data are obtained by quantifying the latent representation matrix $\widetilde{V}^{(t-1)}$ of the old data chunks $\widetilde{X}^{(t-1)}$. Therefore, the auxiliary matrices $\widetilde{Z}_u^{(t-1)}$, $\widetilde{Z}_{x1}^{(t-1)}$ and $\widetilde{Z}_{x2}^{(t-1)}$ need to be updated. $\widetilde{V}_{new}^{(t-1)}$, $\widetilde{Z}_{unew}^{(t-1)}$, $\widetilde{Z}_{x1new}^{(t-1)}$ and $\widetilde{Z}_{x2new}^{(t-1)}$ denote the update of $\widetilde{V}^{(t-1)}$, $\widetilde{Z}_u^{(t-1)}$, $\widetilde{Z}_{x1}^{(t-1)}$ and $\widetilde{Z}_{x2}^{(t-1)}$ under the t-th round of hashing model, respectively. According to the matrix factorization, we have:

$$\widetilde{X}_1^{(t-1)} \approx U_1^{(t)} \widetilde{Z}_{unew}^{(t-1)} \widetilde{L}^{(t-1)} \approx \widetilde{U}_1^{(t-1)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)}, \quad (30)$$

$$\widetilde{X}_2^{(t-1)} \approx U_2^{(t)} \widetilde{Z}_{unew}^{(t-1)} \widetilde{L}^{(t-1)} \approx \widetilde{U}_2^{(t-1)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)}, \quad (31)$$

$$\widetilde{X}_1^{(t-1)} \approx U_{x1}^{(t)} \widetilde{Z}_{x1new}^{(t-1)} \widetilde{L}^{(t-1)} \approx \widetilde{U}_{x1}^{(t-1)} \widetilde{Z}_{x1}^{(t-1)} \widetilde{L}^{(t-1)}, \quad (32)$$

$$\widetilde{X}_2^{(t-1)} \approx U_{x2}^{(t)} \widetilde{Z}_{x2new}^{(t-1)} \widetilde{L}^{(t-1)} \approx \widetilde{U}_{x2}^{(t-1)} \widetilde{Z}_{x2}^{(t-1)} \widetilde{L}^{(t-1)}. \quad (33)$$

According to (30), (31), (32) and (33), $\widetilde{Z}_{unew}^{(t-1)}$, $\widetilde{Z}_{x1new}^{(t-1)}$ and $\widetilde{Z}_{x2new}^{(t-1)}$ can be solved by the following equations:

$$\mathcal{F}_1 = \lambda \left\| U_1^{(t)} \widetilde{Z}_{unew}^{(t-1)} \widetilde{L}^{(t-1)} - \widetilde{U}_1^{(t-1)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + (1-\lambda) \left\| U_2^{(t)} \widetilde{Z}_{unew}^{(t-1)} \widetilde{L}^{(t-1)} - \widetilde{U}_2^{(t-1)} \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \gamma \left\| \widetilde{Z}_{unew}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2, \quad (34)$$

$$\mathcal{F}_2 = \lambda \left\| U_{x1}^{(t)} \widetilde{Z}_{x1new}^{(t-1)} \widetilde{L}^{(t-1)} - \widetilde{U}_{x1}^{(t-1)} \widetilde{Z}_{x1}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \gamma \left\| \widetilde{Z}_{x1new}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2, \quad (35)$$

$$\mathcal{F}_3 = (1-\lambda) \left\| U_{x2}^{(t)} \widetilde{Z}_{x2new}^{(t-1)} \widetilde{L}^{(t-1)} - \widetilde{U}_{x2}^{(t-1)} \widetilde{Z}_{x2}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2 + \gamma \left\| \widetilde{Z}_{x2ew}^{(t-1)} \widetilde{L}^{(t-1)} \right\|_F^2. \quad (36)$$

Let $\frac{\partial \mathcal{F}_1}{\partial \widetilde{Z}_{unew}^{(t-1)}} = 0$, $\frac{\partial \mathcal{F}_2}{\partial \widetilde{Z}_{x1new}^{(t-1)}} = 0$ and $\frac{\partial \mathcal{F}_3}{\partial \widetilde{Z}_{x2new}^{(t-1)}} = 0$, we have

$$\widetilde{Z}_{unew}^{(t-1)} = \left(\lambda U_1^{(t)T} U_1^{(t)} + (1-\lambda)\, U_2^{(t)T} U_2^{(t)} + \gamma I\right)^{-1} \left((\lambda U_1^{(t)T} \widetilde{U}_1^{(t-1)} + (1-\lambda)\, U_2^{(t)T} \widetilde{U}_2^{(t-1)}) \widetilde{Z}_u^{(t-1)} \widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)\left(\widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)^{-1}, \quad (37)$$

$$\widetilde{Z}_{x1new}^{(t-1)} = \left(\lambda U_{x1}^{(t)T} U_{x1}^{(t)} + \gamma I\right)^{-1} \left(\lambda U_{x1}^{(t)T} \widetilde{U}_{x1}^{(t-1)} \widetilde{Z}_{x1}^{(t-1)} \widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)\left(\widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)^{-1}, \quad (38)$$

$$\widetilde{Z}_{x2new}^{(t-1)} = \left((1-\lambda) U_{x2}^{(t)T} U_{x2}^{(t)} + \gamma I\right)^{-1} \left((1-\lambda) U_{x2}^{(t)T} \widetilde{U}_{x2}^{(t-1)} \widetilde{Z}_{x2}^{(t-1)} \widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)\left(\widetilde{L}^{(t-1)} \widetilde{L}^{(t-1)T}\right)^{-1}. \quad (39)$$

It can be observed from (37), (38) and (39) that the updating of the matrices $\widetilde{Z}_{unew}^{(t-1)}$, $\widetilde{Z}_{x1new}^{(t-1)}$ and $\widetilde{Z}_{x2new}^{(t-1)}$ are independent of old data chunks. Therefore, we can efficiently update the latent semantic representation of old data to fit the latest hashing model without accessing the original old data.

3.4 Hash functions learning

In this paper, the hash functions adopt a linear regression model. For the out-of-sample instances, the original image and text features can be linearly projected into the latent semantic space, respectively. To achieve the purpose of online learning, the hash function can be defined as follows:

$$\min_{P_{x1}^{(t)}, P_{x2}^{(t)}} \left\| V^{(t)} - P_{x1}^{(t)} X_1^{(t)} \right\|_F^2 + \left\| V^{(t)} - P_{x2}^{(t)} X_2^{(t)} \right\|_F^2 + \left\| \widetilde{V}^{(t-1)} - P_{x1}^{(t)} \widetilde{X}_1^{(t-1)} \right\|_F^2 + \left\| \widetilde{V}^{(t-1)} - P_{x2}^{(t)} \widetilde{X}_2^{(t-1)} \right\|_F^2 + \gamma \left(\left\| P_{x1}^{(t)} \right\|_F^2 + \left\| P_{x2}^{(t)} \right\|_F^2 \right). \quad (40)$$

Let the derivatives of (40) w.r.t $P_{x1}^{(t)}$ and $P_{x2}^{(t)}$ be equal to zero, respectively. Thus, the solution of $P_{x1}^{(t)}$ is expressed as follows:

$$P_{x1}^{(t)} = F_1^{(t)} \left(D_1^{(t)} + \gamma I\right)^{-1}, \quad (41)$$

where $F_1^{(t)}$ and $D_1^{(t)}$ are expressed as follows:

$$\begin{aligned} F_1^{(t)} &= V^{(t)} X_1^{(t)T} + \widetilde{V}^{(t-1)} \widetilde{X}_1^{(t-1)T} \\ &= V^{(t)} X_1^{(t)T} + \widetilde{F}_1^{(t-1)}, \end{aligned} \tag{42}$$

$$\begin{aligned} D_1^{(t)} &= X_1^{(t)} X_1^{(t)T} + \widetilde{X}_1^{(t-1)} \widetilde{X}_1^{(t-1)T} \\ &= X_1^{(t)} X_1^{(t)T} + \widetilde{D}_1^{(t-1)}. \end{aligned} \tag{43}$$

And the solution of $P_{x2}^{(t)}$ is given as follows:

$$P_{x2}^{(t)} = F_2^{(t)} \left(D_2^{(t)} + \gamma I \right)^{-1}, \tag{44}$$

where $F_2^{(t)}$ and $D_2^{(t)}$ are expressed as follows:

$$\begin{aligned} F_2^{(t)} &= V^{(t)} X_2^{(t)T} + \widetilde{V}^{(t-1)} \widetilde{X}_2^{(t-1)T} \\ &= V^{(t)} X_2^{(t)T} + \widetilde{F}_2^{(t-1)}, \end{aligned} \tag{45}$$

$$\begin{aligned} D_2^{(t)} &= X_2^{(t)} X_2^{(t)T} + \widetilde{X}_2^{(t-1)} \widetilde{X}_2^{(t-1)T} \\ &= X_2^{(t)} X_2^{(t)T} + \widetilde{D}_2^{(t-1)}. \end{aligned} \tag{46}$$

The hash codes of the query data can be obtained directly by the projection matrix.

In summary, Algorithm 1 details the solution steps of the OSCMFH method.

Input: Data chunks $\left[X^{(1)}, X^{(2)}, \ldots, X^{(t)}\right]$ with label $\left[L^{(1)}, L^{(2)}, \ldots, L^{(t)}\right]$, parameters λ, μ and γ, hash codes length k and the maximum number of iterations T.
Output: Hash codes B, projection matrix P_{x1} and P_{x2}.
1. **Initialize** $U_1^{(0)}$, $U_{x1}^{(0)}$, $U_{x2}^{(0)}$, $Z_u^{(0)}$, $Z_{x1}^{(0)}$, $Z_{x2}^{(0)}$, and $W^{(0)}$ by random matrices.
2. **Initialize** $E_m^{(0)}$, $C_m^{(0)}$, $F^{(0)}$, $F_1^{(0)}$, $F_2^{(0)}$, $D^{(0)}$, $D_1^{(0)}$, and $D_2^{(0)}$ by zero matrices, $m = 1, 2, 3, 4$.
3. **for** $chunk = 1 \rightarrow t$
(1) Update $D_1^{(t)}$ and $D_2^{(t)}$ by (43) and (46).
Repeat
(2) Update $E_1^{(t)}$, $C_1^{(t)}$ and $U_1^{(t)}$ by (8), (9) and (7).
(3) Update $E_2^{(t)}$, $C_2^{(t)}$ and $U_2^{(t)}$ by (11), (12) and (10).
(4) Update $E_3^{(t)}$, $C_3^{(t)}$ and $U_{x1}^{(t)}$ by (15), (16) and (14).
(5) Update $E_4^{(t)}$, $C_4^{(t)}$ and $U_{x2}^{(t)}$ by (18), (19) and (17).
(6) Update $F^{(t)}$, $D^{(t)}$ and $W^{(t)}$ by (22), (23) and (21).
(7) Update $Z_u^{(t)}$, $Z_{x1}^{(t)}$ and $Z_{x2}^{(t)}$ by (25), (27) and (28).
(8) Update $V^{(t)}$ by (29).
Until convergence or reaching the maximum iteration.
if $chunk \geq 2$ **then**
(9) Update $\widetilde{Z}_{unew}^{(t-1)}$, $\widetilde{Z}_{x1new}^{(t-1)}$ and $\widetilde{Z}_{x2new}^{(t-1)}$ by (37), (38) and (39).
(10) Set $\widetilde{V}_{new}^{(t-1)} = \begin{bmatrix} \widetilde{Z}_{unew}^{(t-1)} \ \widetilde{L}^{(t-1)} \\ \widetilde{Z}_{x2new}^{(t-1)} \ \widetilde{L}^{(t-1)} \end{bmatrix}$.
(11) Set $V = [\widetilde{V}_{new}^{(t-1)}, V^{(t)}]$.
end if
(12) Update $F_1^{(t)}$ and $F_2^{(t)}$ by (42) and (45).
(13) Update $P_{x1}^{(t)}$ and $P_{x2}^{(t)}$ by (41) and (44).
end for
4. Set $B = sgn(V)$, $P_{x1} = P_{x1}^{(t)}$ and $P_{x2} = P_{x2}^{(t)}$.

Algorithm 1 OSCMFH.

3.5 Complexity analysis

In this section, we give the computational complexity analysis of the proposed OSCMFH approach. In the t-th round, we need to update the relevant variables matrix of the newly arrived data to obtain the latent semantic representation $V^{(t)}$. Meanwhile, the parameter matrix related to the latent semantic representation $\widetilde{V}^{(t-1)}$ of the old data chunk needs to be updated in this round. Finally, hash codes are obtained by quantifying the combination of latent semantic representations of old and new data. Due to d_1, d_2, k_1, k_2 and $c \ll N$, let $m = \max\{d_1, d_2, k_1, k_2, c\}$. Therefore, the time complexity of the training phase and the query phase in the t-th round are $o(m^2NT)$ and $o(m^2N)$, respectively, where T is the number of iterations in round t. Obviously, the time complexity of the OSCMFH method is linear with the training set size N.

4 Experiments

In this section, extensive experiments on four benchmark datasets were conducted to verify the effectiveness of OSCMFH. In addition, we compared the proposed method with several online cross-modal hashing methods. The source code of our proposed OSCMFH method will be publicly available later.

4.1 Datasets

To simulate the streaming data for online learning, the training set of each dataset is divided into multiple data chunks. Table 1 shows the details of the training sets of the four datasets, which were divided into chunks of different sizes.

Wiki [34]: 2866 image-text pairs of the Wiki dataset were collected from Wikipedia, which was divided into 10 different categories. Each image and text can be described by a 128-dimensional SIFT feature vector and a 10-dimensional feature vector, respectively. We randomly selected 693 image-text pairs as the retrieval set and the remaining 2173 image-text pairs as the training set.

Table 1 The details of the training set division

Training set division	Wiki	MIRFlickr	NUS-WIDE	MSCOCO
Training set size	2173	15902	184710	18000
Number of data chunks	11	8	37	9
Data chunk size except the last	200	2000	5000	2000
Last data chunk size	173	1902	4710	2000

MIRFlickr [35]: The MIRFlickr dataset consists of 25,000 image-text pairs from 24 different categories. As the setting in [36], we can selected 16,738 pairs as the experiment dataset. Each sample of image modality is represented by a 150-dimensional edge histogram vector, and each data point of text modality is represented by a 500-dimensional feature vector representation. 836 image-text pairs were randomly chosen as the retrieval set and the remaining 15902 image-text pairs as the training set.

NUS-WIDE [37]: In NUS-WIDE dataset, 269,548 image-text pairs were collected from Flickr. Since some classes contain few samples, we chose the top 10 common classes. Therefore, 186577 image-text pairs were used as the experimental dataset. Specially, we randomly selected 1867 image-text pairs as the retrieval set and the remaining 184710 image-text pairs as the training set. Each image and text can be represented by a 500-dimensional bag-of-words vector and a 1000-dimensional feature vector, respectively.

MSCOCO [38]: 122,218 image-text pairs of the MSCOCO dataset were collected from Flickr with 80 different categories. In the experiments, each image is represented as a 4096-dimensional deep feature extracted by the Caffe implementation of the VGG network [39], and each text is presented as a 2000-dimensional feature using the bag-of-words model. We randomly picked out 7762 image-text pairs as the retrieval set and the remaining 18000 image-text pairs as the training set.

4.2 Evaluation metrics

We employ the mean Average Precision (mAP) to evaluate the retrieval performance of each method. Here, mAP is defined as follows:

$$mAP = \frac{1}{Q}\sum_{q=1}^{Q}\frac{1}{N}\sum_{r=1}^{R}P_q(r)\delta_q(r) \tag{47}$$

where Q is the query instance, and N is the number of relevant instances in the retrieval set. In our experiment, R is the total number of retrieved data points and it is set to 100. $P_q(r)$ is the precision of the top-r retrieved instances of the q-th query. If the q-th query instance is related to the r-th instance, then $\delta_q(r) = 1$, otherwise $\delta_q(r) = 0$. In general, the larger the mAP value, the better the retrieval performance.

4.3 Comparison methods

We compare the proposed OSCMFH method with six state-of-the-art online CMH retrieval methods, such as OCMH [7], OCMFH [8], OLSH [9], FOMH [11], OASH [31] and DOCH [32]. Among these online learning methods, both OCMH and OCMFH are completely unsupervised hashing methods. The rest methods, such as OLSH, FOMH, OASH, and DOCH are supervised methods by considering the label information or similarity between samples.

4.4 Experimental settings

There are three parameters λ, μ, and γ in the OSCMFH method. In the following experiments, we set λ, μ and γ to 0.1, 0.01, and 0.01, respectively. And k_1 is set to three quarters of hash codes length. In addition, the maximum number of iterations per round is set to 40. Among them, the codes of OCMFH, FOMH and DOCH are published by the author, and the codes of other methods are reproduced by us. All methods were run five times, and the final average results were reported.

4.5 Experimental results

The mAP values of OSCMFH and other comparison methods on Wiki, MIRFlickr, NUS-WIDE, and MSCOCO datasets with different hash code lengths are shown in Tables 2, 3, 4, and 5, respectively. From the experimental results on four cross-modal datasets, we can get the following observations:

1) As shown in Table 2, compared with other datasets, all methods achieve relatively low retrieval performance on the Wiki dataset in the image retrieval text task. One of the reasons may be that the semantic gap between image

Table 2 mAP values on Wiki dataset

Task	Methods	Hash codes length(bits)			
		16	32	64	128
Image	OCMH	0.1654	0.1659	0.1666	0.1665
	OCMFH	0.2232	0.2268	0.2326	0.2182
	OLSH	0.2286	0.2393	0.2445	0.2467
to	FOMH	0.1927	0.2011	0.2192	0.2221
	OASH	0.2290	0.2378	0.2432	0.2510
Text	DOCH	**0.2419**	**0.2486**	0.2312	0.2178
	OSCMFH	0.2382	0.2432	**0.2456**	**0.2541**
Text	OCMH	0.1964	0.1822	0.1766	0.1981
	OCMFH	0.5751	0.5893	0.5951	0.5663
	OLSH	0.5502	0.5760	0.5984	0.6076
to	FOMH	0.5321	0.5758	0.5901	0.5957
	OASH	0.6467	0.6544	0.6554	0.6563
Image	DOCH	**0.6537**	0.6539	0.6465	0.6413
	OSCMFH	0.6471	**0.6664**	**0.6707**	**0.6735**

The bold entries indicates the best performance

modality and text modality in the Wiki dataset is relatively large, which leads to lower retrieval performance than in other datasets. Furthermore, the performances of most methods in the task of image retrieval text are inferior to that in the task of text retrieval image. The possible reason is that the samples from the image modality lose less information when they are mapped to hash codes. In addition, textual information can characterize more semantic information than visual features from the semantic point of view.

2) It can be found that OASH outperforms other comparison methods in most cases. It indicates that the hash codes of new and old data can be updated when the new data arrives, and thus it achieves a better match between new and old data. It can be seen from Table 5 that our proposed OSCMFH method cannot outperform OASH on the MSCOCO dataset when the lengths of hash codes are 16bits and 32bits. Specifically, the mAP values of OASH vary between 0.8253 and 0.9457 by setting the hash codes from 16bits to

Table 3 mAP values on MIRFlickr dataset

Task	Methods	Hash codes length(bits)			
		16	32	64	128
Image	OCMH	0.6009	0.6055	0.6013	0.6011
	OCMFH	0.6440	0.6436	0.6360	0.6326
	OLSH	0.6478	0.6556	0.6738	0.6783
to	FOMH	0.6557	0.6856	0.7035	0.7122
	OASH	0.7600	0.7791	0.7869	0.7901
Text	DOCH	0.7457	0.7591	0.7564	0.7561
	OSCMFH	**0.7878**	**0.8161**	**0.8258**	**0.8348**
Text	OCMH	0.5923	0.5905	0.5907	0.5905
	OCMFH	0.7119	0.7270	0.7517	0.7718
	OLSH	0.6684	0.7059	0.7476	0.7804
to	FOMH	0.7086	0.7735	0.8039	0.8169
	OASH	0.8770	0.9024	0.9184	0.9180
Image	DOCH	0.8615	0.8756	0.8869	0.8875
	OSCMFH	**0.9135**	**0.9281**	**0.9336**	**0.9336**

The bold entries indicates the best performance

Table 4 mAP values on NUS-WIDE dataset

Task	Methods	Hash codes length(bits)			
		16	32	64	128
Image	OCMH	0.4666	0.4720	0.4723	0.4702
	OCMFH	0.4351	0.4634	0.4662	0.4656
	OLSH	0.5394	0.5600	0.5888	0.5995
to	FOMH	0.5128	0.5387	0.5322	0.5611
	OASH	0.7322	0.7587	0.7749	0.7784
Text	DOCH	0.7329	0.7076	0.7082	0.7074
	OSCMFH	**0.7753**	**0.8081**	**0.8107**	**0.8281**
Text	OCMH	0.4534	0.4552	0.4492	0.4518
	OCMFH	0.5256	0.5668	0.5864	0.5941
	OLSH	0.5976	0.6509	0.7075	0.7348
to	FOMH	0.5941	0.6249	0.6567	0.7022
	OASH	0.8562	0.8678	0.8691	0.8742
Image	DOCH	0.8516	0.8502	0.8589	0.8564
	OSCMFH	**0.8875**	**0.8997**	**0.8963**	**0.9020**

The bold entries indicates the best performance

128bits. The variation range of our proposed OSCMFH method is between 0.7950 and 0.9604. The possible reason is that the learned hash codes of OSCMFH are more sensitive to the change of hash code length on the MSCOCO dataset. Therefore, our OSCMFH method cannot achieve better performances than OASH on this dataset in some cases. In addition, it is easy to know that our OSCMFH method achieves the best retrieval performance with different hash codes length among all methods in most cases. It verifies the effectiveness of our proposed OSCMFH approach in cross-modal retrieval tasks. The main reason may be that the OSCMFH method not only fully utilizes the labels of the multimodality data but also takes into account the specific properties of each modality.

3) It is clear to see that the performances of most of the retrieval methods increase with the increase of the length of hash codes. This is because longer hash

Table 5 mAP values on MSCOCO dataset

Task	Methods	Hash codes length(bits)			
		16	32	64	128
Image	OCMH	0.4125	0.4050	0.4072	0.4253
	OCMFH	0.4392	0.4343	0.4347	0.4374
	OLSH	0.4153	0.4198	0.4235	0.4256
to	FOMH	0.4148	0.4508	0.4615	0.4653
	OASH	0.4501	0.5076	0.4978	0.5039
Text	DOCH	0.3945	0.3956	0.4105	0.4132
	OSCMFH	**0.5363**	**0.5529**	**0.5575**	**0.5590**
Text	OCMH	0.4501	0.4514	0.4413	0.4465
	OCMFH	0.4468	0.4443	0.4446	0.4455
	OLSH	0.4264	0.4364	0.4578	0.5055
to	FOMH	0.4585	0.5823	0.6360	0.6632
	OASH	**0.8253**	**0.9043**	0.9344	0.9457
Image	DOCH	0.7570	0.8613	0.9303	0.9479
	OSCMFH	0.7950	0.8896	**0.9461**	**0.9604**

The bold entries indicates the best performance

codes can usually preserve more semantic information of multimodality data. However, with the increase in hash code length, the retrieval performances of the algorithms decrease in some cases. The main reasons may be that the longer hash codes preserve more similarity among the samples, but they also increase redundant information. Therefore, these methods may be unsuitable for the long hash codes length on these datasets.

To evaluate the online learning strategy, we conducted some retrieval experiments on MIRFlickr and NUS-WIDE datasets with hash code lengths of 16, 32, and 64, respectively. Figures 2 and 3 show the mAP curves of

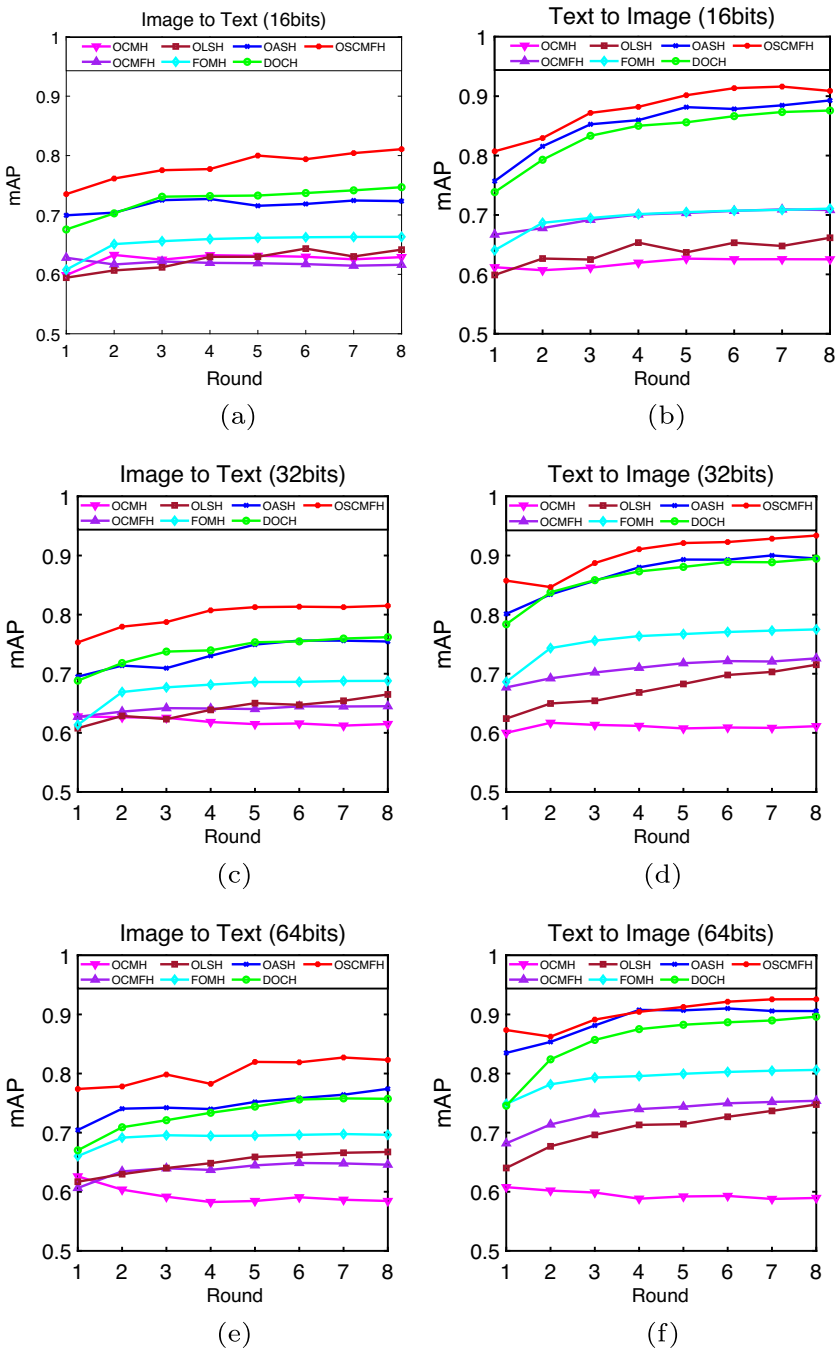


Fig. 2 The mAP values in each round on MIRFlickr dataset

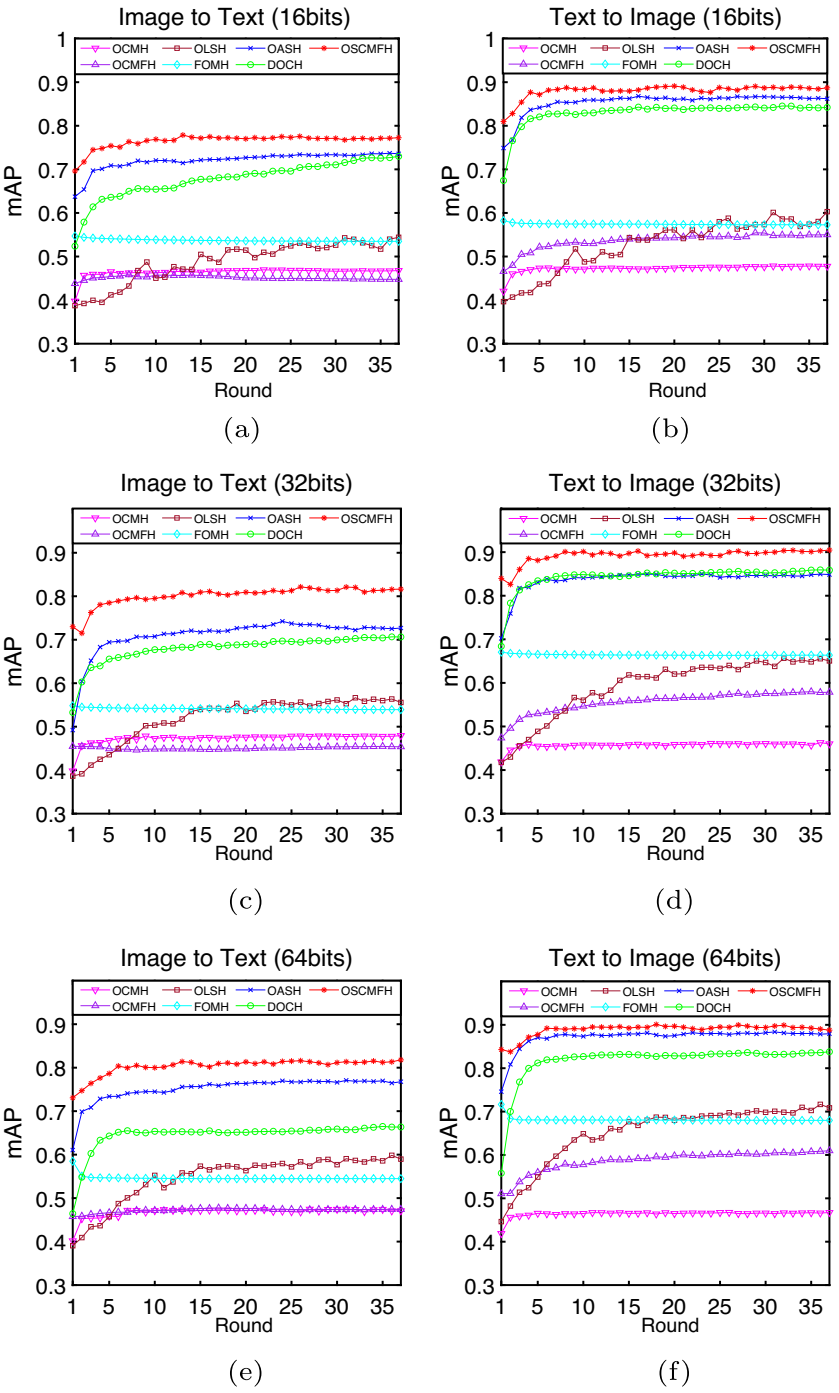


Fig. 3 The mAP values in each round on NUS-WIDE dataset

different algorithms on the MIRFlickr and NUS-WIDE datasets, respectively. From the results, we can observe that the mAP values of almost all methods increase with the addition of new data, which indicates that online hashing methods can deal with large-scale streaming data. Moreover, our proposed OSCMFH method outperforms other competitors on MIRFlickr and NUS-WIDE datasets with different hash code lengths, which verifies the effectiveness of the online cross-modal retrieval approaches.

4.6 Time cost analysis

To verify the efficiency of online hashing methods, we have conducted some experiments to compare the training cost of our proposed online hashing method with its corresponding offline version on the MIRFlickr dataset. When new data arrives, the online hashing methods do not need to access the previous old data, while the offline hashing methods need to retrain the new and old data. Table 6 records the training

Table 6 Training time (seconds) for online and offline methods on MIRFlickr

Data chunk	16bits		32bits		64bits		128bits	
	Online	Offline	Online	Offline	Online	Offline	Online	Offline
1th	2.97	2.97	3.53	3.53	4.14	4.14	5.41	5.41
2th	1.50	4.98	1.78	5.95	2.58	8.53	4.19	14.07
3th	1.52	7.02	1.88	8.76	2.61	13.31	4.22	20.88
4th	1.50	8.92	1.81	11.17	2.70	15.76	4.30	27.36
5th	1.52	11.32	1.77	13.76	2.50	20.11	4.20	32.89
6th	1.48	13.41	1.82	16.16	2.55	24.36	4.19	39.63
7th	1.52	15.27	1.89	18.67	2.59	27.84	4.55	46.59
8th	1.53	17.08	1.73	22.67	2.53	32.19	4.02	51.31

time cost of the online hashing and its offline version. It can be seen from Table 6 that the more data chunks, the online hashing method takes less computational cost than its offline hashing version. This also shows the superiority of online hashing in terms of training efficiency.

4.7 Ablation experiment

To verify the contributions of modality-specific properties and the update of the latent semantic correlation matrix of old data when new data arrives in OSCMFH, we conducted ablation experiments on the MIRFlickr dataset to verify its effectiveness. We constructed two variants of OSCMFH, named OSCMFH-S and OSCMFH-U. OSCMFH-S removes the relevant loss terms specific to the modality attributes. Therefore, only the loss terms related to the common semantic correlation matrix are retained in OSCMFH-S. OSCMFH-U indicates that the latent semantic correlation matrix in the t round is not updated in the $t+1$ round. Therefore, the latent semantic correlation matrix of the old data in OSCMHF-U is not updated when the new data arrive. The experimental results are shown in Table 7.

It can be seen from Table 7 that our proposed OSCMFH method outperforms OSCMFH-S on the MIRFlickr dataset. It manifests that the modality-specific properties can be preserved by learning modality-specific latent semantic representations. Therefore, the learned hash codes of OSCMHF have stronger discrimination ability. In addition, we can observe that our OSCMFH method also achieves better retrieval accuracy than OSCMFH-U on the MIRFlickr dataset. The main reason is that OSCMFH can update the latent semantic representation of old data as new data arrives to fit the latest hashing model.

4.8 Parameter sensitivity analysis

In this section, we carried out experiments on four benchmark datasets to analyze the parameter sensitivity by setting the hash code length to 32 bits. We vary one parameter by fixing other parameters. Figure 4 shows the mAP values of all methods with the different values of the parameters λ, μ, and γ. From the results in Fig. 4, we can obtain the following observations:

1) We can see that the mAP values of our approach on four datasets are relatively stable while the parameter λ is in the range of [0.0001, 0.9]. In addition, when the value of the parameter λ is greater than 0.9, the retrieval accuracy in the text-to-image task decreases, especially

Table 7 The MAP results of various methods on MIRFlickr dataset

Task	Methods	16bits	32bits	64bits	128bits
Image	OSCMFH-S	0.7807	0.7899	0.8268	0.8201
to	OSCMFH-U	0.7693	0.7979	0.7925	0.7846
Text	OSCMFH	0.7826	0.8124	0.8306	0.8311
Text	OSCMFH-S	0.9071	0.9194	0.9195	0.9234
to	OSCMFH-U	0.9029	0.9215	0.9197	0.9164
Image	OSCMFH	0.9118	0.9227	0.9327	0.9306

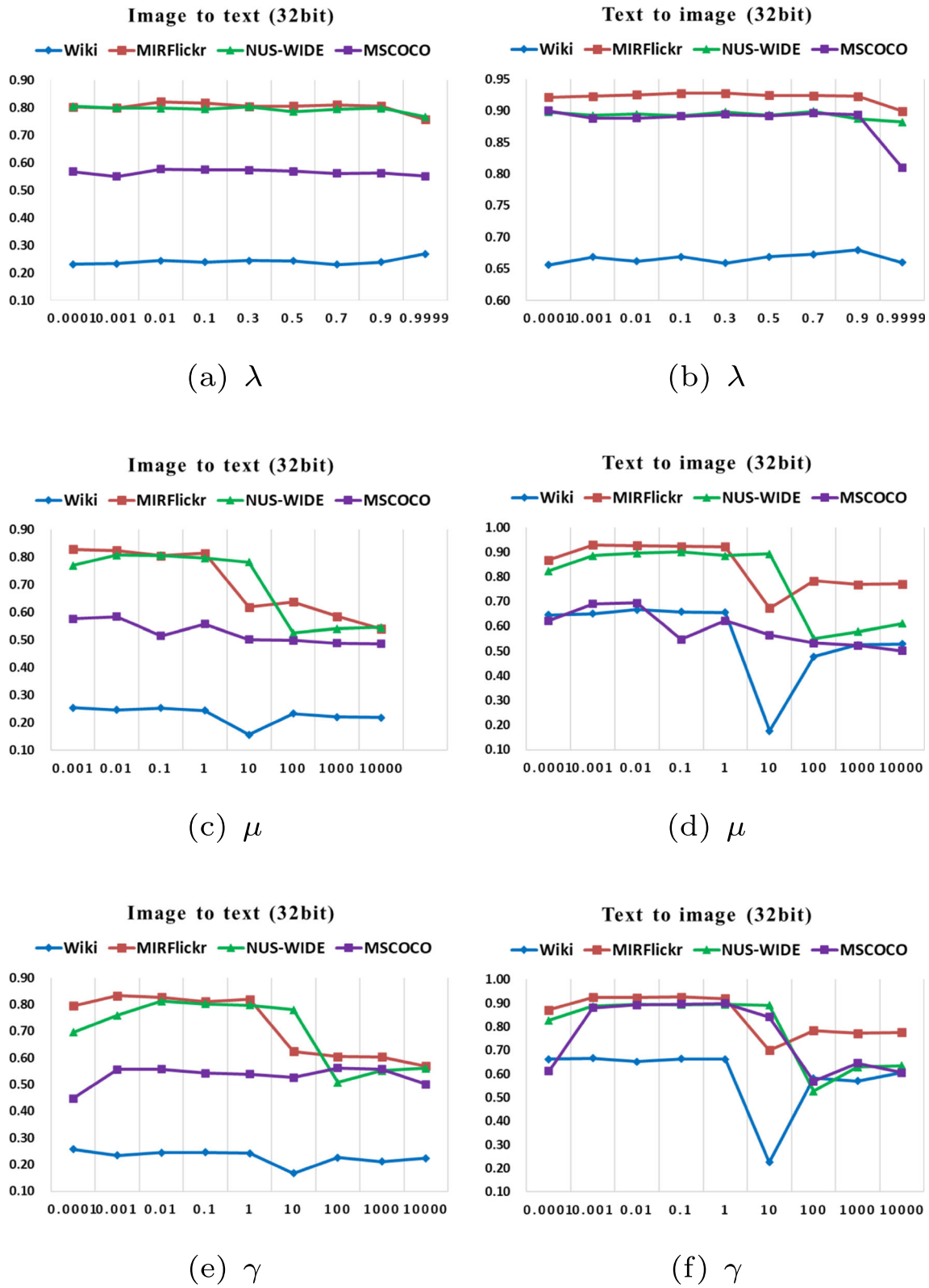


Fig. 4 Sensitivity analysis of λ, μ and γ

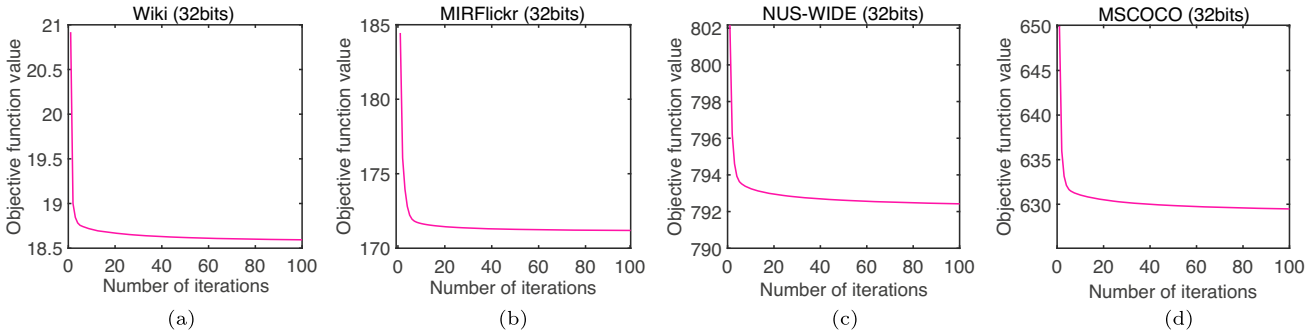


Fig. 5 Convergence curves on different datasets

on the dataset MSCOCO. Therefore, the optimal value of the parameter λ is $[0.0001,\ 0.9]$.

2) When the values of the parameters μ and γ are in the range of $[0.001,\ 0.1]$, OSCMFH can obtain relatively stable retrieval performance. Once the values of μ and γ are larger than 1, the mAP values of the proposed approach on the four datasets decrease significantly, especially in the task of text querying images. Thus, the optimal values of μ and γ can be in the range of $[0.001,\ 0.1]$.

4.9 Convergence analysis

Since the proposed model adopts an online iterative updating strategy, its convergence rate is crucial to the retrieval performance. To verify the convergence of our OSCMFH method, we conducted some experiments on four datasets. Here, the hash codes length of our method was set to 32 bits in this experiment. Figure 5 shows the convergence curves of the OSCMFH method on the first chunk of data on the four datasets. It can be seen that the OSCMFH method converges within 20 iterations on all datasets and its convergence speed is very fast, which verifies the efficiency of the OSCMFH method. Furthermore, we conducted some experiments to verify the relationship between the number of iterations and the retrieval performance of our proposed method. Figure 6 shows the performances of our OSCMFH method in the first four rounds on the MIRFlickr dataset. It can be seen that our proposed OSCMFH method can converge within ten iterations in the first round. Furthermore, we can know that our method converges within five iterations from the second round. This demonstrates the fast convergence of the proposed OSCMFH method.

5 Conclusions

In this paper, we propose an online supervised collective matrix factorization hashing(OSCMFH) method for large-scale data stream retrieval. Our OSCMFH method preserves the shared and specific properties of multimodal data by decomposing each modality into shared and specific semantic representations. In addition, the semantic labels are fully utilized to improve retrieval performance. Since our OSCMFH method is an online learning method, the latent semantic representation of old data can be updated without accessing the old data. Compared with other online hashing learning methods, the learned hash codes of the proposed OSCMFH method can effectively capture the semantic information, and thus show powerful discriminative ability. The experiment results on four benchmark datasets have validated the superiority of the OSCMFH method in cross-modal retrieval tasks.

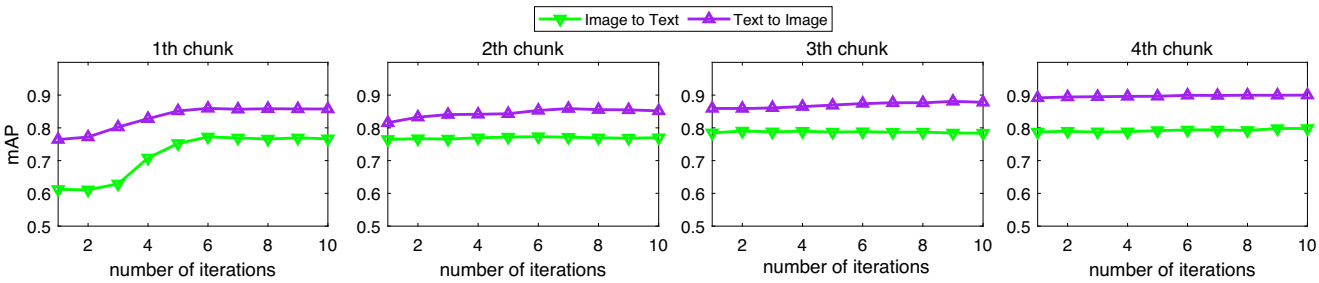


Fig. 6 The effect of the number of iterations on retrieval performance

Acknowledgements This work was supported by the National Natural Science Foundation of China [Grant No. 61603159, 62162033, U21B2027, U1836218], Yunnan Provincial Major Science and Technology Special Plan Projects [Grant No. 202002AD080001, 202103AA080015], Yunnan Foundation Research Projects [Grant No. 202201AT070154, 202101BE070001-056].

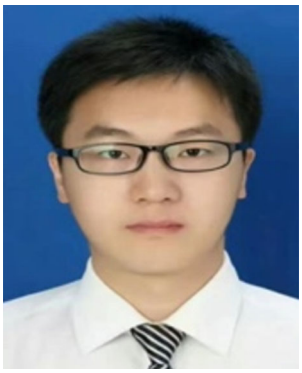
References

1. Yu Ju, Wu X-J, Kittler J (2019) Discriminative supervised hashing for cross-modal similarity search. Image Vis Comput 89:50–56
2. Zhong F, Chen Z, Min G, Xia F (2020) A novel strategy to balance the results of cross-modal hashing. Pattern Recogn 107:107523
3. Shen HT, Liu L, Yang Y, Xu X, Zi H, Shen F, Hong R (2020) Exploiting subspace relation in semantic labels for cross-modal hashing. IEEE Trans Knowl Data Eng 33(10):3351–3365
4. Shu Z, Yong K, Yu J, Gao S, Mao C, Yu Z (2022) Discrete asymmetric zeroshot hashing with application to cross-modal retrieval. Neurocomputing 511:366–379
5. Yu J, Zhang D, Shu Z, Chen F (2022) Adaptive multi-modal fusion hashing via hadamard matrix. Appl Intell:1–15
6. Shu Z, Bai Y, Zhang D, Yu J, Yu Z, Wu X-J (2022) Specific class center guided deep hashing for cross-modal retrieval. Inf Sci 609:304–318
7. Xie L, Shen J, Zhu L (2016) Online cross-modal hashing for web image retrieval. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 30
8. Di W, Wang Q, An Y, Gao X, Tian Y (2020) Online collective matrix factorization hashing for large-scale cross-media retrieval. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 1409–1418
9. Yao T, Wang G, Yan L, Kong X, Su Q, Zhang C, Qi T (2019) Online latent semantic hashing for cross-media retrieval. Pattern Recogn 89:1–11
10. Yi J, Liu X, Cheung Y-M, Xu X, Fan W, Yi He (2021) Efficient online label consistent hashing for large-scale cross-modal retrieval. In: 2021 IEEE internationa conference on multimedia and expo (ICME) IEEE, pp 1–6.
11. Lu X, Zhu L, Cheng Z, Li J, Nie X, Zhang H (2019) Flexible online multi-modal hashing for large-scale multimedia retrieval. In: Proceedings of the 27th ACM international conference on multimedia, pp 1129–1137
12. Di W, Wang Q, He L, Gao X, Tian Y (2020) Joint and individual matrix factorization hashing for large-scale cross-modal retrieval. Pattern Recogn 107:107479
13. Lu W, Yang J, Zareapoor M, Zheng Z (2021) Cluster-wise unsupervised hashing for cross-modal similarity search. Pattern Recogn 111:107732
14. Fang X, Liu Z, Na H, Jiang L, Teng S (2021) Discrete matrix factorization hashing for cross-modal retrieval. Int J Mach Learn Cybern 12(10):3023–3036
15. Yao T, Li Y, Guan W, Wang G, Li Y, Yan L, Tian Q (2021) Discrete robust matrix factorization hashing for large-scale cross-media retrieval. IEEE Trans Knowl Data Eng
16. Li H, Zhang C, Jia X, Gao Y, Chen C (2021) Adaptive label correlation based asymmetric discrete hashing for cross-modal retrieval. IEEE Trans Knowl Data Eng
17. Zhang D, Wu X-J, Xu T, Yin H (2021) Dah: discrete asymmetric hashing for efficient cross-media retrieval. IEEE Trans Knowl Data Eng
18. Zhang D, Wu X-J, Yin H-F, Moon JK (2021) Multi-hash codes joint learning for cross-media retrieval. Pattern Recogn Lett 151:19–25
19. Di W, Gao X, Wang X, He L (2018) Label consistent matrix factorization hashing for large-scale cross-modal similarity search. IEEE Trans Pattern Anal Mach Intell 41(10):2466–2479
20. Liu X, Hu Z, Ling H, Mtfh Y-MC (2019) A matrix tri-factorization hashing framework for efficient cross-modal retrieval. IEEE Trans Pattern Anal Mach Intell 43(3):964–981
21. Cakir F, Bargal SA, Sclaroff S (2017) Online supervised hashing. Comput Vis Image Underst 156:162–173
22. Cakir F, He K, Bargal SA, Mihash SS (2017) Online hashing with mutual information. In: Proceedings of the IEEE international conference on computer vision, pp 437–445
23. Cakir F, Sclaroff S (2015) Adaptive hashing for fast similarity search. In: Proceedings of the IEEE international conference on computer vision, pp 1044–1052
24. Huang L-K, Yang Q, Zheng W (2013) Online hashing. In: IJCAI, pp 1422–1428
25. Leng C, Wu J, Cheng J, Bai X, Lu H (2015) Online sketching hashing. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2503–2511
26. Lin M, Ji R, Liu H, Sun X, Wu Y, Wu Y (2019) Towards optimal discrete online hashing with balanced similarity. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 8722–8729
27. Lin M, Ji R, Liu H, Wu Y (2018) Supervised online hashing via hadamard codebook learning. In: Proceedings of the 26th ACM international conference on multimedia, pp 1635–1643
28. Lin M, Ji R, Sun X, Zhang B, Huang F, Tian Y, Tao D (2020) Fast class-wise updating for online hashing. IEEE Trans Pattern Anal Mach Intell
29. Lin M, Ji R, Liu H, Sun X, Chen S, Qi T (2020) Hadamard matrix guided online hashing. Int J Comput Vis 128(8):2279–2306
30. Lin M, Ji R, Chen S, Sun X, Lin C-W (2020) Similarity-preserving linkage hashing for online image retrieval. IEEE Trans Image Process 29:5289–5300
31. Su R, Di W, Huang Z, Liu Y, An Y (2020) Online adaptive supervised hashing for large-scale cross-modal retrieval. IEEE Access 8:206360–206370
32. Zhan Y-W, Wang Y, Yu S, Wu X-M, Luo X, Xu X-S (2022) Discrete online cross-modal hashing. Pattern Recogn 122:108262
33. Lu X, Zhu L, Cheng Z, Nie L, Zhang H (2019) Online multi-modal hashing with dynamic query-adaption. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, pp 715–724
34. Rasiwasia N, Pereira JC, Coviello E, Doyle G, Lanckriet GRG, Levy R, Vasconcelos N (2010) A new approach to cross-modal multimedia retrieval. In: Proceedings of the 18th ACM international conference on multimedia, pp 251–260
35. Huiskes MJ, Lew MS (2008) The mir flickr retrieval evaluation. In: Proceedings of the 1st ACM international conference on multimedia information retrieval, pp 39–43
36. Lin Z, Ding G, Hu M, Wang J (2015) Semantics-preserving hashing for cross-view retrieval. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3864–3872
37. Chua T-S, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) Nus-wide: a real-world web image database from national university of singapore. In: Proceedings of the ACM international conference on image and video retrieval, pp 1–9

38. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, pp 740–755
39. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556

Zhenqiu Shu received the Ph.D. degree in computer applications at Nanjing University of Science and Technology. In February 2021, he joined the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, where he is currently an associate professor. Before joining in Kunming University of Science and Technology University, he had been a postdoctoral in Jiangnan University for four years. His research interests include image processing, computer vision and machine learning.

Li Li is currently pursuing toward Master degree at the Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. Her current research interests include multimedia information retrieval and machine learning.



Jun Yu received his Ph.D. degree at the school of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He joined the College of Computer and Communication Engineering, Zhengzhou University of Light Industry in 2021. His research interests include multimedia information retrieval, computer vision and deep learning.

Donglin Zhang received the Ph.D. degree at the school of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, China. He is currently a lecturer in the school of artificial intelligence and computer science of Jiangnan University. His current research interests include multimedia information retrieval and pattern recognition.

Zhengtao Yu received the Ph.D. degree in computer application technology from the Beijing Institute of Technology, Beijing, China, in 2005. He is currently a Professor with the School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China. His current research interests include natural language processing, information retrieval, and machine learning.

Xiao-Jun Wu received the B.Sc. degree in mathematics from Nanjing Normal University, Nanjing, China, in 1991, and the M.S. and Ph.D. degrees in pattern recognition and intelligent systems from the Nanjing University of Science and Technology, Nanjing, in 1996 and 2002, respectively. From 1996 to 2006, he taught at the School of Electronics and Information, Jiangsu University of Science and Technology, where he was promoted to Professor. He has been with the School of AI & CS, Jiangnan University, since 2006, where he is a Professor of Computer Science and Technology. He was a Visiting Researcher with the CVSSP, University of Surrey, U.K., from 2003 to 2004. He has published over 300 research papers in refereed international journals and conferences. His current research interests include pattern recognition and computational intelligence. He was a Fellow of the International Institute for Software Technology, United Nations University, from 1999 to 2000. He was a recipient of the Most Outstanding Postgraduate Award from the Nanjing University of Science and Technology.