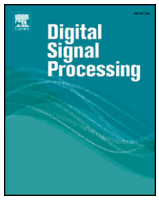
Digital Signal Processing 143 (2023) 104226

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp



Adaptive semi-paired query hashing for multi-modal retrieval

Jun Yu [a], Wei Huang [a], Zuhe Li [a], Kunlin Li [a], Zhenqiu Shu [b,*], Qi Wang [c], Jinge Jiao [c]

[a] The College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, China
[b] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, China
[c] Zhengzhou Branch Company of China Unicom, Zhengzhou, China

ARTICLE INFO

Article history:
Available online 23 September 2023

Keywords:
Hashing
Multi-modal retrieval
Semi-paired query
Cross-modal reconstruction
Batch hash coding

ABSTRACT

Multi-modal hashing has attracted enormous attention in large-scale multimedia retrieval, owing to its advantages of low storage cost and fast Hamming distance computation. Existing multi-modal hashing methods assume that all multi-modal data are well paired and then encode the paired multiple modalities into joint binary codes. However, it is not ensured that all data are fully paired in practical applications. In this paper, we present an adaptive semi-paired query hashing method, which facilitates learning the hash codes for semi-paired query samples. The proposed method performs projection learning and cross-modal reconstruction learning to maintain the semantic consistency between multi-modal data. Meanwhile, the semantic similarity structure and the complementary multi-modal information are preserved by hash codes to obtain a discriminative hash function. In the encoding stage, the missing modality features are completed via the learned cross-modal reconstruction matrices. In addition, the multimodal fusion weights are fine-tuned adaptively for the new query data to capture the modality difference. The extensive experiment results on three benchmark datasets show that our proposed algorithm outperforms state-of-the-art multi-modal hashing methods.

1. Introduction

Nowadays, it is a universal truth that information search encounters challenges with the tremendous amounts of multimedia data continuing to grow at an astonishing speed in social networks [20]. Hashing, as an effective representation technique, plays an important role in information retrieval, recommendation, and computer vision due to its advantages of low storage and high-efficiency [28]. The goal of hashing methods is to map high-dimensional data into a string of compact binary codes. Thereby, the approximate nearest neighbor (ANN) search can be accelerated by fast Exclusive OR (XOR) operation in large-scale data. Early pioneers focused on image search and proposed various uni-modal hashing methods [7,4,23,8,10,15,14,9,13]. Uni-modal hashing is widely applied to image search applications and achieved satisfactory performance.

However, uni-modal hashing works on a single modality and is intractable for multi-modal applications. In practical object search tasks, a target object usually is characterized by multiple modalities from different aspects. For example, an image in WeChat moments is often attached by related text descriptions. There is an emerging need that supports similarity measures across different modalities in the field of information retrieval. Cross-modal hashing methods [29,30,34,37,6,12,35,31,26] are proposed to implement the search task where taking one modality as a query to retrieve the other modalities from a database. Nevertheless, uni-modal hashing and cross-modal hashing fail to generate directly the joint hash codes for paired multi-modal data.

Recently, with the success of multimodal technology in recognition tasks [1,2], some researchers have focused on learning joint hash codes for multi-view features. An intuitive way is to concatenate multiple modality features as the input of uni-modal hashing models. However, the processing way causes information redundancy and a curse of dimensionality. To overcome the problem, multi-modal hashing methods, such as Multiple Feature Hashing (MFH) [27], Multi-view Latent Hashing (MVLH) [25], Multi-view Alignment Hashing (MAH) [16], Deep Multi-modal Hashing (SID-MH) [17], and Hadmard matrix-Guided Multi-modal Hashing (HGMH) [33], are studied to code heterogeneous multi-modal data into a string of binary codes. These methods exploit the complementarity between multi-modal data to learn a joint hash function that can generate hash codes for query data with paired modalities. Besides, some online multi-modal hashing works [18,36] are developed to dynamically capture the modality difference of new queries. However, the joint hash function learned by the above

* Corresponding author.
E-mail addresses: yujun@zzuli.edu.cn (J. Yu), hnhw235@163.com (W. Huang), zuheli@123.com (Z. Li), 1125050701@qq.com (K. Li), szqkunming@163.com (Z. Shu), 295162570@qq.com (Q. Wang), 809898461@qq.com (J. Jiao).

https://doi.org/10.1016/j.dsp.2023.104226
1051-2004/

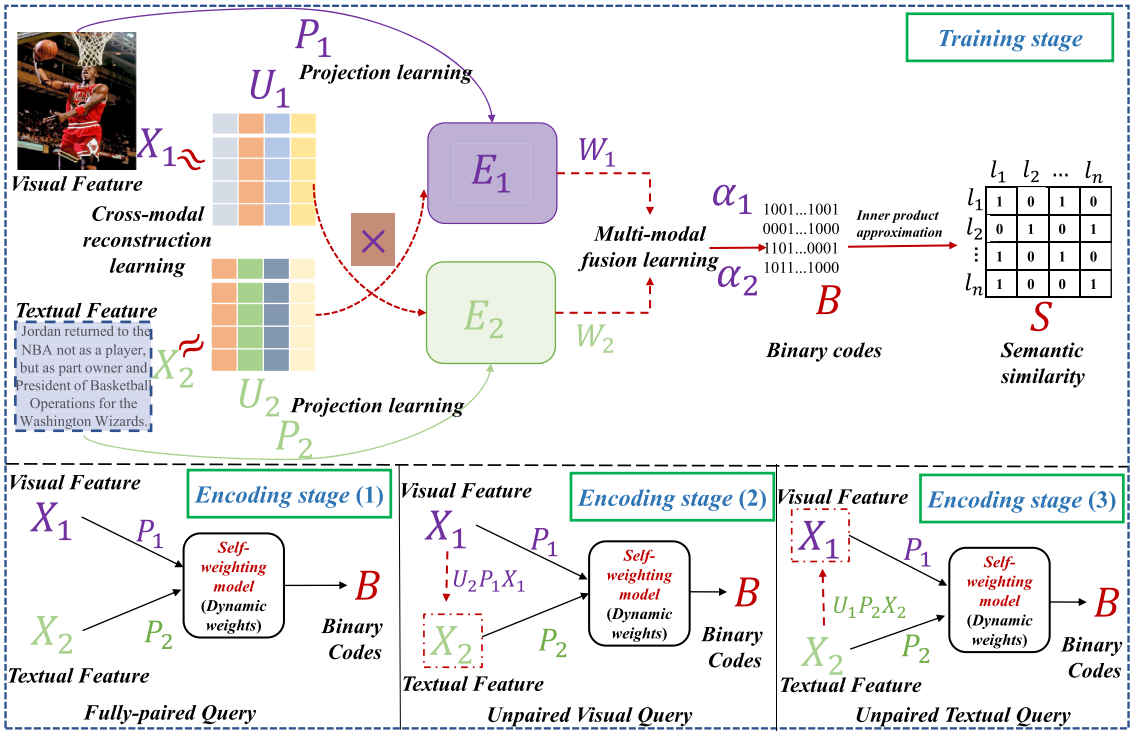


Fig. 1. The framework of our proposed ASPQH. Our framework is divided into two stages, i.e., the training stage and the encoding stage. In the training stage, the multi-modal data is first projected into a low-dimensional subspace. In the subspace, one modality can be reconstructed by the subspace representation of any other modality. Then, the multi-modal information is fused and the semantic structure in label space is preserved by hash codes in Hamming space. In the encoding stage, we employ the learned cross-modal reconstruction matrices to complete the missing modality features and then adopt the self-weighting binary learning method to generate hash codes of new queries.

methods is not applied to unpaired query samples. It is not sure that all modalities are completed in real applications. To address this problem, a novel multi-modal hashing method called adaptively semi-paired query hashing (ASPQH) is proposed in this paper to learn the hash codes for semi-paired query data. Fig. 1 illustrates the flowchart of our proposed ASPQH. The proposed framework includes an offline training stage and an online encoding stage. In the offline training stage, we jointly perform projection learning, cross-modal reconstruction learning, multi-modal fusion learning, and semantic preservation hashing learning to learn discriminative hash function. In the encoding stage, our model cleverly uses the learned cross-modal reconstruction matrices to complete the missing modality. Considering the modality information difference between training data and testing data, we adopt the dynamic weighting mode to obtain the hash codes of new queries. The main contributions of our proposed ASPQH are summarized as follows.

- We develop an adaptively semi-paired query hashing framework that jointly performs subspace learning and hashing learning. Our learning architecture fuses the complementary multi-modal information and preserves the semantic structure information in label space by hash codes.
- We design cross-modal reconstruction terms to enhance the semantic consistency representation across heterogeneous modalities. Our model employs the learned reconstruction matrices to solve effectively the joint code problem for unpaired samples.
- An effective optimization method is proposed to solve our objective function with a discrete constraint. A comparative evaluation of our proposed method with state-of-the-art methods on three datasets shows the effectiveness of our proposed method.

Structurally, the rest of this paper is organized as follows. Section 2 reviews the related works of hash learning. The proposed adaptive sem-paired query hashing model is described in Section 3. In Section 4, we present and discuss the experimental results on three datasets. The conclusions of the paper are drawn in Section 5.

2. Related work

In this section, we preliminarily review hashing-based retrieval methods. The existing hashing methods can be classified into three branches, i.e., uni-modal hashing, cross-modal hashing, and multi-modal hashing.

2.1. Uni-modal hashing

Uni-modal hashing is designed to implement hash retrieval on a single modality. Uni-modal hashing methods are divided into data-independent methods and data-dependent ones. The representative data-independent methods include Locality-sensitive Hashing (LSH) [4], Kernelized Locality Sensitive Hashing (KLSH) [11], Shift-invariant Kernel Hashing (SIKH) [21], etc. The data-independent methods need a long code length to maintain high performance, which raises memory consumption and storage cost. Data-dependent methods usually have better retrieval accuracy than data-independent methods. Iterative Quantization (ITQ) [7] is a classic data-dependent hashing method. It uses an orthogonal rotation matrix to reduce the quantization loss. Supervised Discrete Hashing (SDH) [23] develops an effective discrete optimization method to solve the discrete variable without any relaxation operations. Discrete Locality Linear Embedding Hashing (DLLH) [8] preserves the locality manifold structure of original data space by hash codes. An anchor-based acceleration scheme is proposed to extend DLLH to large-scale data scenarios. Hadamard Codebook

based Online Hashing (HCOH) [15] associates those data sharing the same label with a codeword sampled from the Hadamard matrices to train the hash function online.

2.2. Cross-modal hashing

Cross-modal hashing aims at retrieving other relevant modalities by a given modality as the query. The learned hash functions can encode each modality separately but can not directly encode paired multiple modalities to obtain their joint hash codes. Recently, various cross-modal hashing methods are grouped into supervised cross-modal hashing and unsupervised cross-modal hashing. Unsupervised cross-modal hashing tends to learn hash functions by exploring data distribution and topological structure. Joint and Individual Matrix Factorization Hashing (JIMFH) [30] learns unified hash codes and individual hash codes to preserve the shared properties of multi-modal data and the specific properties of each modality, respectively. Unsupervised Multi-modal Hashing (UMH) [35] explores both the semantic correlation of textual modality and the local geometric structure of visual modality in a hashing learning framework. Supervised cross-modal hashing methods employ the semantic labels of training data to learn compact hash codes. Such methods can achieve higher retrieval performance than unsupervised ones. Supervised Robust Discrete Multimodal Hashing (SRDMH) [12] learns discriminative hash functions based on label information and preserves the similarity of the original space. Semi-supervised and Semi-paired Hashing (SSH) [34] is proposed to learn the robust hash functions under semi-paired and semi-supervised multi-modal data situations. Semantic-enhanced Discrete Matrix Factorization Hashing (SDMFH) [6] learns a common hashing representation of all modalities by factorizing the intermodal semantic similarity graph. Scalable Asymmetric Discrete Cross-modal Hashing (BATCH) [31] preserves the pairwise label distances through hash codes and improves algorithm complexity. Scalable Discrete Matrix factorization and Semantic Autoencoder (SDMSA) [37] reconstructs original modality features via common binary features to enhance the representation capacity of hash codes. Specific Class Center Guided Deep Hashing (SCCGDH) [26] learns specific class centers under a neural network architecture. Similar to uni-modal hashing, cross-modal hashing takes one modality as the query to be coded.

2.3. Multi-modal hashing

Multi-modal hashing learns a joint hash function to encode the paired multi-modal data, different from cross-modal hashing and uni-modal hashing. Most existing multi-modal hashing approaches try to fuse multi-modal information in an unsupervised learning manner. For example, Multiple Feature Hashing (MFH) [27] first constructs multi-view graphs and a global graph, and then preserves these structure information in hash learning. Besides, Multi-view Latent Hashing (MVLH) [25] adopts the matrix factorization in the kernel feature space to learn compact and unified binary codes. Multiview Discrete Hashing (MVDH) [24] develops a non-negative spectral clustering method to obtain the cluster pseudo labels, then performs the collective matrix factorization and mapping learning to keep the consistency between the cluster labels and hash codes. Semantic-driven Interpretable Deep Multi-modal Hashing (SIDMH) [17] designs a deep hashing network to generate interpretable hash codes. The output of each branch of the SIDMH model is concatenated to obtain the hash codes of any query sample with category information. Hadamard Matrix-Guided Multi-modal Hashing (HGMH) [33] jointly performs the subspace learning and target feature learning to learn a discriminative and unified hash function.

Although the above methods have achieved great progress, the modality difference between different multi-modal data can be further studied, which may provide potential improvement. The hardened hash function learned from the training stage is not applied to all queries. Therefore, Multi-modal Hashing methods with dynamic query code are developed in recent years. Online Multi-modal Hashing with Dynamic Query-adaption (OMH-DQ) [18] exploits the complementarity of multi-modal information and the supervision of pair-wise semantic labels to learn a robust joint hash function. Adaptive Multi-modal Fusion Hashing (AMFH) [36] introduces the Hadamard matrix to generate hash centers for specific classes and induces the samples with the same class to be close to their common class center point in Hamming space. In the above methods, the hash codes of query samples are generated with the adaptive modality weights to capture the modality difference among multi-modal data.

However, the aforementioned multi-modal hashing methods need to ensure that the query sample has complete modal information to obtain its joint hash codes. Currently, few multi-modal hashing methods provide solutions to generate binary codes for unpaired query data. In this paper, we propose a novel Adaptive Semi-paired Query Hashing (ASPQH) to learn a discriminative and joint hash function and apply it to encode semi-paired multi-modal data.

3. The proposed method

3.1. Model formulation

Assume that a training dataset consists of n image-text pairs. The image set and the text set are represented as $Y^{(1)} = [y_1^{(1)}, \cdots, y_n^{(1)}]$ and $Y^{(2)} = [y_1^{(2)}, \cdots, y_n^{(2)}]$, respectively, where $y_i^{(1)} \in R^{d_x}$ and $y_i^{(2)} \in R^{d_y}$. The label matrix is $L = [l_1, \cdots, l_n] \in R^{r \times n}$, where r denotes the number of categories. $l_i = [l_{i1}, \cdots, l_{ir}]^T \in R^r$, where $l_{ij} = 1$ if the i-th object is classified into the j-th category, otherwise $l_{ij} = 0$. We adopt the radial basis function (RBF) kernel to obtain the image features and text features, for example, the i-th sample of the m-th modality is calculated by $X_m^i = [exp(\frac{\|y_i^{(m)} - A_1^{(m)}\|^2}{-2\sigma_m^2}), \cdots, exp(\frac{\|y_i^{(m)} - A_k^{(m)}\|^2}{-2\sigma_m^2})]^T$, where m is equal to 1 or 2 and σ_m is the kernel width of the m-th modality; $A^{(1)} \in R^{d_x \times k}$ and $A^{(2)} \in R^{d_y \times k}$ are the anchor samples that are taken randomly from image modality and text modality, respectively. In this paper, we use X_1 and X_2 to denote the feature matrix of image modality and text modality, respectively.

The paired different modalities describe the same semantic content. Therefore, the consistency semantically can be viewed as a bridge across different modalities. The original modality features of one modality can be reconstructed by the subspace representation of other any modality to capture the highly semantic correlation between the paired different modalities. We aim to find a latent subspace to achieve the goal and the "cross-modal reconstruction" idea is defined as the following formula:

$$\begin{aligned} \min_{P_1, P_2, E_1, E_2, U_1, U_2} & \lambda(\|P_1 X_1 - E_1\|_F^2 + \|X_1 - U_1 E_2\|_F^2) \\ & + (1-\lambda)(\|P_2 X_2 - E_2\|_F^2 \\ & + \|X_2 - U_2 E_1\|_F^2) \end{aligned} \tag{1}$$

where λ is a hyperparameter that is manually tuned within a range of [0 1] to balance the projection learning and cross-modal reconstruction terms of image modality and text modality. $P_m \in R^{d \times k}$ and $U_m \in R^{k \times d}$ are the mapping matrix and the basis matrix of the m-th modality, respectively. $E_m \in R^{d \times n}$ denotes the feature representation of the m-th modality in the latent subspace. "Cross" in

cross-modal reconstruction learning module is embodied by image feature reconstruction term $\|X_1 - U_1E_2\|_F^2$ and text feature reconstruction term $\|X_2 - U_2E_1\|_F^2$ in Eq. (1). The purpose of our model is to fuse complementary multi-modal information to learn joint hash codes. The semantic structure information in label space is preserved by hash codes to improve the discriminative ability of our model and boost retrieval performance. Specifically, the problem is formulated as follows.

$$\begin{aligned}\min_{W_1,W_2,B,\alpha_m} &\sum_{m=1}^{M}(\alpha_m)^t\|W_mE_m - B\|_F^2 + \beta\|S - B^TB\|_F^2 \\ s.t.&B \in \{+1,-1\}^{c\times n}, \sum_{m=1}^{M}\alpha_m = 1, \alpha_m > 0,\end{aligned} \tag{2}$$

where $W_m \in R^{c\times d}$ represents the mapping matrix that transforms the features of the m-th modality to Hamming space. It cost $\mathcal{O}(n^2)$ to store $S \in R^{n\times n}$ and is infeasible when the number of the training data is quite large. In this paper, we replace the affinity matrix S with the inner product L^TL in the optimization, where S is defined as $S_{ij} = \frac{l_il_j}{\|l_i\|_2\|l_j\|_2}$. α_m denotes the weighting coefficient of the m-th modality and t is a smoothing parameter. The overall optimization problem is formulated as follows.

$$\begin{aligned}\min_{\substack{P_1,P_2,E_1,E_2,U_1,\\ U_2,W_1,W_2,B,\alpha_m}} &\lambda(\|P_1X_1 - E_1\|_F^2 + \|X_1 - U_1E_2\|_F^2) \\ &+ (1-\lambda)(\|P_2X_2 - E_2\|_F^2 + \|X_2 - U_2E_1\|_F^2) \\ &+ \sum_{m=1}^{M}(\alpha_m)^t\|W_mE_m - B\|_F^2 + \beta\|S - B^TB\|_F^2 \\ &+ \gamma\Phi(P_1,P_2,U_1,U_2,W_1,W_2) \\ s.t.&B \in \{+1,-1\}^{c\times n}, \sum_{m=1}^{M}\alpha_m = 1, \alpha_m > 0,\end{aligned} \tag{3}$$

where λ, β and γ are balance parameters. In objective function (3), P_1, P_2, W_1, W_2, U_1, U_2, α_1 and α_2 are the necessary parameters of the joint hash function to be learned.

3.2. Optimization method

It is difficult directly to solve the non-convex optimization problem (3). In this subsection, an iterative algorithm is proposed to solve any one of multiple variables with other variables fixed. The updating rules are summarized in the following steps.

1) Step 1: Fixing other variables and rewriting the objective function of P_1, we can obtain

$$\min_{P_1}\lambda\|P_1X_1 - E_1\|_F^2 + \gamma\|P_1\|_F^2 \tag{4}$$

Letting the derivative of (4) concerning P_1 to zero, we obtain the closed-form solution of P_1.

$$P_1 = \lambda E_1X_1^T(\lambda X_1X_1^T + \gamma I)^{-1} \tag{5}$$

2) Step 2: Fixing other variables and rewriting the objective function of P_2, we can obtain

$$\min_{P_2}\lambda\|P_2X_2 - E_2\|_F^2 + \gamma\|P_2\|_F^2 \tag{6}$$

Letting the derivative of (6) concerning P_2 to zero, we obtain the closed-form solution of P_2.

$$P_2 = \lambda E_2X_2^T(\lambda X_2X_2^T + \gamma I)^{-1} \tag{7}$$

3) Step 3: Fixing other variables and rewriting the objective function of E_1, we obtain

$$\begin{aligned}\min_{E_1}&\lambda\|P_1X_1 - E_1\|_F^2 + (1-\lambda)\|X_2 - U_2E_1\|_F^2 \\ &+ (\alpha_1)^t\|W_1E_1 - B\|_F^2\end{aligned} \tag{8}$$

Letting the derivative of (8) concerning E_1 to zero, we can derive

$$\begin{aligned}E_1 =&(\lambda I + (1-\lambda)U_2^TU_2 + \alpha_1^tW_1^TW_1)^{-1} \\ &(\lambda P_1X_1 + (1-\lambda)U_2^TX_2 + \alpha_1^tW_1^TB)\end{aligned} \tag{9}$$

4) Step 4: Fixing other variables and rewriting the objective function of E_2, we obtain

$$\begin{aligned}\min_{E_2}&(1-\lambda)\|P_2X_2 - E_2\|_F^2 + \lambda\|X_1 - U_1E_2\|_F^2 \\ &+ (\alpha_2)^t\|W_2E_2 - B\|_F^2\end{aligned} \tag{10}$$

Letting the derivative of (10) concerning E_2 to zero, we can derive

$$\begin{aligned}E_2 =&((1-\lambda)I + \lambda U_1^TU_1 + \alpha_2^tW_2^TW_2)^{-1} \\ &((1-\lambda)P_2X_2 + \lambda U_1^TX_1 + \alpha_2^tW_2^TB)\end{aligned} \tag{11}$$

5) Step 5: Fixing other variables and rewriting the objective function of U_1, we can get

$$\min_{U_1}\lambda\|X_1 - U_1E_2\|_F^2 + \gamma\|U_1\|_F^2 \tag{12}$$

Letting the derivative of (12) concerning U_1 to zero, we derive

$$U_1 = \lambda X_1E_2^T(\lambda E_2E_2^T + \gamma I)^{-1} \tag{13}$$

6) Step 6: Fixing other variables and rewriting the objective function of U_2, we can get

$$\min_{U_2}(1-\lambda)\|X_2 - U_2E_1\|_F^2 + \gamma\|U_2\|_F^2 \tag{14}$$

Letting the derivative of (14) concerning U_2 to zero, we derive

$$U_2 = (1-\lambda)X_2E_1^T((1-\lambda)E_1E_1^T + \gamma I) \tag{15}$$

7) Step 7: Fixing other variables and rewriting the objective function of W_m, we obtain

$$\min_{W_m}(\alpha_m)^t\sum_{m=1}^{M}\|W_mE_m - B\|_F^2 + \gamma\sum_{m=1}^{M}\|W_m\|_F^2 \tag{16}$$

Letting the derivative of (16) concerning W_m to zero, we can derive

$$W_1 = (\alpha_1)^tBE_1^T((\alpha_1)^tE_1E_1^T + \gamma I)^{-1} \tag{17}$$

$$W_2 = (\alpha_2)^tBE_2^T((\alpha_2)^tE_2E_2^T + \gamma I)^{-1} \tag{18}$$

8) Step 8: Fixing other variables and rewriting the objective function of α_m, we can get

$$\begin{aligned}&\min_{\alpha_m}\sum_{m=1}^{M}G_m \\ &s.t.\sum_{m=1}^{M}\alpha_m = 1, \alpha_m > 0\end{aligned} \tag{19}$$

where $G_m = \|W_mE_m - B\|_F^2$. The optimal α_m is obtained by optimizing the Lagrange function of (19).

$$\alpha_m = \frac{(G_m)^{\frac{1}{(1-t)}}}{\sum_{m=1}^{M}(G_m)^{\frac{1}{(1-t)}}} \tag{20}$$

9) Step 9: Fixing other variables and rewriting the objective function of B, we can obtain

$$\min_{\alpha_m} \sum_{m=1}^{M} \|W_m E_m - B\|_F^2 + \beta \|B^T B - S\|_F^2 \quad s.t. B \in \{+1, -1\}^{c \times n} \tag{21}$$

The objective function Eq. (21) with a binary constraint is an NP-hard problem. We introduce an auxiliary variable V to approximate one of B variables and get an equivalent form:

$$\min_{B,V} \sum_{m=1}^{M} \|W_m E_m - B\|_F^2 + \beta \|B^T V - S\|_F^2 + \rho \|B - V\|_F^2 \quad s.t. B \in \{+1, -1\}^{c \times n} \tag{22}$$

where ρ is a balance parameter. We adopt an alternative optimization way to solve B and V. Specifically, when fixing V, we obtain the closed form solution

$$B = sign(\sum_{m=1}^{M} (\alpha_m)^t W_m E_m + \beta V S^T + \rho V) \tag{23}$$

When fixing B, we let the derivative of the objective function (22) concerning V to zero, then

$$V = (\beta B B^T + \rho I)^{-1} (\beta B S + \rho B) \tag{24}$$

In the training stage, the above steps are repeated until the overall objective function converges or the number of iterations reaches the preset maximum value. The optimization steps are summarized in Algorithm 1. After the above optimization process, the optimal P_1, P_2, U_1, U_2, W_1 and W_2 are given to the following stages.

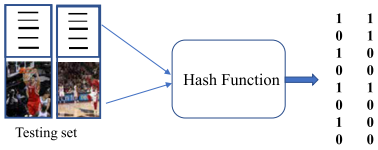
Algorithm 1 The learning process of hash function.

Input: Training set $O = \{(Y_1^1, Y_2^1, l_1), ..., (Y_1^n, Y_2^n, l_n)\}$.
Calculating the non-linear representation matrices: X_1, X_2.
Initializing U_1, U_2, E_1, E_2, W_1, W_2, B, α_1, α_2.
repeat
  Update P_1 according to Eq. (5);
  Update P_2 according to Eq. (7);
  Update E_1 according to Eq. (9);
  Update E_2 according to Eq. (11);
  Update U_1 according to Eq. (13);
  Update U_2 according to (15);
  Update W_1 according to (17);
  Update W_2 according to (18);
  Update $\alpha_m (m = 1, ..., M)$ according to (20);
  Update B according to (23) and (24);
until convergence
Output: P_1, P_2, U_1, U_2, W_1, W_2, α_1 and α_2.

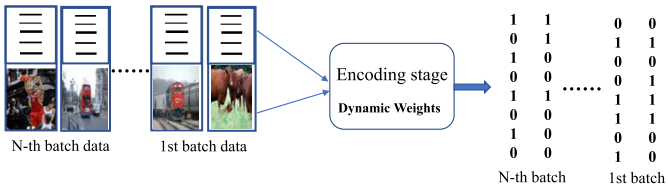
3.3. Unpaired query extension

Most of the existing multi-modal hashing methods assume that all data are well paired. However, there are unpaired query samples in many applications. In this subsection, we employ the learned U_1 and U_2 in the training stage to cope with the unpaired query data. Specifically, if the image features of the query data q are not available, we can complete them based on the available text subspace features E_2 and the construction matrix U_1. We have

$$\min_{X_1^q} \|X_1^q - U_1 E_2\|_F^2 \tag{25}$$



(a) Conventional encoding process



(b) The online batch encoding process

Fig. 2. Schematic diagram of the difference between traditional encoding method and online encoding method in the encoding process.

where $E_2 = P_2 X_2^q$. We can easily acquire image features

$$X_1^q = U_1 P_2 X_2^q \tag{26}$$

Similarly, if the query data q only contains image modality features, that is, the text modality features are missing. The missing text features can be calculated according to the following formula:

$$\min_{X_2^q} \|X_2^q - U_2 E_1\|_F^2 \tag{27}$$

where $E_1 = P_1 X_1^q$. We have text features

$$X_2^q = U_2 P_1 X_1^q \tag{28}$$

As far, we have obtained the paired image feature matrix X_1^q and text feature matrix X_2^q of a query set.

3.4. Encoding stage

The previous multi-modal methods usually applied the learned joint hash function $f(X_1^q, X_2^q) = sign(\alpha_1 W_1 P_1 X_1^q + \alpha_2 W_2 P_2 X_2^q)$ in the training stage to obtain the joint hash codes of paired multiple modalities. The weighting coefficient of each modality is fixed in the encoding stage. However, there is a large modality difference between different multi-modal data. The weighting coefficients learned from the training stage do not adapt to all query data. Therefore, the weights of different modalities should be adjusted dynamically with an endless stream of data. Fig. 2 illustrates the difference between conventional offline encoding and online encoding. Assume that X_m^q denotes the m-th modality of the newly arrived query data, the learning problem is defined as follows.

$$\min_{\alpha_m, B_q} \sum_{m=1}^{M} (\alpha_m^q)^t \|Q_m X_m^q - B_q\|_F^2 \quad s.t. \sum_{m=1}^{M} \alpha_m^q = 1, B \in \{+1, -1\}^{c \times n} \tag{29}$$

where $Q_m = W_m P_m$. The Eq. (29) can be solved alternately according to the following two steps. Concretely,

Step1: Optimization of α_m^q

$$\alpha_m^q = \frac{(G_m^q)^{\frac{1}{(1-t)}}}{\sum_{m=1}^{M} (G_m^q)^{\frac{1}{(1-t)}}} \tag{30}$$

Step2: Optimization of B

$$B_q = sign(\sum_{m=1}^{M} (\alpha_m^q)^t Q_m X_m^q) \tag{31}$$

As seen in Algorithm 2, the above process is performed iteratively until (29) converges. The optimal B_q is regarded as the joint hash codes of the query data q.

Algorithm 2 Adaptive Semi-paired Query Hashing (ASPQH).

Input: P_1, P_2, W_1, W_2, X_1^q, X_2^q
If image modality is missing, then X_1^q is obtained according to Eq. (26);
If text modality is missing, then X_2^q is obtained according to Eq. (28);
Calculating $Q_1 = P_1 W_1$ and $Q_2 = P_2 W_2$.
Initializing B_q.
1: **repeat**
2: Update α_1 and α_2 according to Eq. (30);
3: Update B_q according to Eq. (31);
4: **until** convergence
Output: B_q, α_1, α_2.

3.5. Complexity analysis

The objective function (3) is minimized by alternatively updating each variable. In this subsection, we detail the computational complexity of our updating algorithm in the training stage. Suppose that n is the size of the training set, c denotes the hash code length, d and k represent the dimension of the latent subspace and the original kernel space, respectively. In Eq. (5) and (7), the matrix inversion and the matrix multiplication operations require $\mathcal{O}(k^3)$ and $\mathcal{O}(dnk)$, respectively, thus the time complexity of updating P_1 and P_2 is $\mathcal{O}(dnk + k^3)$. Similarly, updating E_1, E_2, U_1 and U_2 cost $\mathcal{O}(dnk)$. In Eq. (17) and (18), it needs $\mathcal{O}(nd^2)$ to update W_1 and W_2 since $d < n$. When updating B, the time complexity is $\mathcal{O}(n^2c)$ if we use directly S to calculate in optimization. In this paper, we utilize label matrix $L^T L$ to replace S, thus the computational complexity is reduced to $\mathcal{O}(cdn)$. Suppose the algorithm converges after ξ iterations, the overall training cost is $\mathcal{O}(\xi(dkn + k^3))$.

4. Experiments

To validate the effectiveness of our proposed method, we conduct comparison experiments with several state-of-the-art methods on three benchmark datasets, i.e., Pascal Sentence[1] [22], NUS-WIDE[2] [3], and IAPR TC-12[3][5]. The detailed statistics of these datasets are presented in Table 1. In experiments, the query data to be encoded in batch is divided into a testing set and a retrieval set. We obtain hash codes batch by batch according to the procedures of the encoding stage. The multi-modal retrieval experiments are performed on three datasets, using the testing set to search related results from the retrieval set.

4.1. Data sets

Pascal Sentence consists of 1,000 image-text pairs. These pairs are divided into 20 categories. Each image is attached in 5 separate sentences. We use the CNN visual features [32] and the 100-dimensional probability vector to represent images and texts, respectively. For a fair comparison, we randomly take 800 image-text pairs to compose a training dataset (40 pairs per class), and the remaining 200 image-text pairs are regarded as a testing dataset (5 pairs per class).

Table 1
Statistics of three benchmark datasets.

DataSets	Pascal Sentence	NUS-WIDE	IAPR TC-12
Data Set Size	1,000	186,577	20,000
Training Set Size	800	5,000	5,000
Retrieval Set Size	800	186,577	18,000
Testing Set Size	200	1,866	2,000
Num. of Labels	20	10	255

NUS-WIDE is comprised of 269,648 image-text samples of 81 concepts. In our experiments, we only keep 186,577 samples of the top ten most frequent concepts. Each image and each text are represented by a 500-dimensional bag-of-visual words vector and the 1000-dimensional tag occurrence vector, respectively. A random subset of 1,866 samples is used as a testing set and the remaining 18,4711 samples are regarded as a retrieval set. We take randomly 5,000 samples from the retrieval set to form a training set.

IAPR TC-12 contains 20,000 image-text samples that are annotated by 255 tags. The dataset is divided into a testing set of 2,000 samples and a database of 18,000 samples. We randomly take a subset of 5,000 samples from the database to form a training set. The text modality of each sample is represented by a 2912-dimensional bag-of-words vector. The 512-dimensional GIST feature vector is extracted to represent image modality.

4.2. Experiment settings

In our experiments, we compare the proposed ASPQH with some hashing models including ITQ [7], LSH [4], DLLH [8], HCOH [15], MFH [27], MVLH [25], OMH-DQ [18], SIDMH [17] and AMFH [36], based on the following evaluation metrics.

The Mean Average Precision (mAP) is one evaluation criterion of retrieval performance. The Average Precision (AP) for given a query q is defined as follows.

$$\mathrm{AP}(q) = \frac{1}{l_q} \sum_{m=1}^{R} P_q(m)\delta_q(m) \tag{32}$$

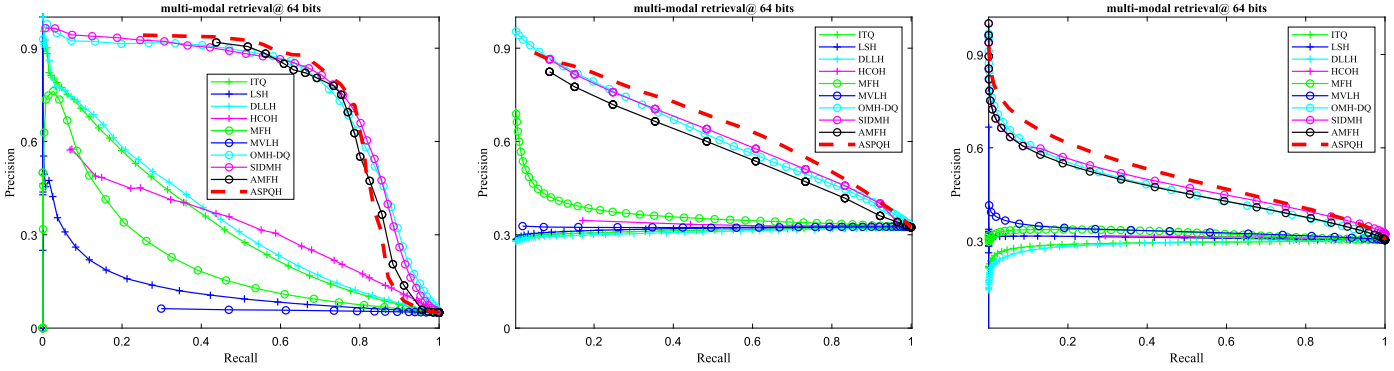
where l_q denotes the number of correct instances in the top R retrieval results; $P_q(m)$ is the accuracy of top m retrieved results; $\delta_q(m) = 1$ if the m-th position is right, otherwise, $\delta_q(m) = 0$. The average AP value of all queries is mAP. The larger mAP indicates the better performance. R is set to the size of the retrieve set in the following experiments.

Among baselines, ITQ, LSH, DLLH, and HCOH are representative uni-modal hashing methods while the others are multi-modal hashing. The multiple modality features are concatenated as the input of all uni-modal methods for a fair comparison. The hyper-parameters in the baseline methods are set according to the suggestions reported in their original papers. In the encoding stage, we empirically set the size of each batch as $0.02 * num$, where num is equal to the number of the training set. In experiments, we tune each hyper-parameter in sequence within a wide range (see Section 4.5) and find an optimal value for our proposed method by fixing other parameters. The detailed parameter settings are shown as follows. $\lambda = 0.5, \beta = 1e^5, \gamma = 1e^{-3}, \rho = 1e^3$ on Pascal Sentence dataset; $\lambda = 0.1, \beta = 1e^1, \gamma = 1e^{-3}, \rho = 1e^3$ on NUS-WIDE dataset; $\lambda = 0.9, \beta = 1e^3, \gamma = 1e^{-5}, \rho = 1e^{-5}$ on IAPR TC-12 dataset. All baseline methods and our proposed ASPQH are implemented on a workstation with MATLAB R2018b. This workstation is configured with 16 GB memory and Intel(R) Core(TM) i7-10700 CPU @2.90 GHz.

[1] https://vision.cs.uiuc.edu/pascal-sentences/.
[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.
[3] https://www.imageclef.org/photodata.

Table 2
mAP comparison under different bits in fully-paired scenarios.

Task	Methods	Pascal Sentence				NUS-WIDE				IAPR TC-12			
		16	32	64	128	16	32	64	128	16	32	64	128
O2O	ITQ	0.3602	0.3523	0.3675	0.3803	0.3724	0.3751	0.3776	0.3789	0.3730	0.3844	0.3936	0.4020
	LSH	0.1011	0.1243	0.1572	0.2129	0.3421	0.3554	0.3544	0.3672	0.3251	0.3363	0.3509	0.3686
	DLLH	0.3631	0.3720	0.3971	0.3959	0.3738	0.3782	0.3794	0.3823	0.3644	0.3796	0.3863	0.3868
	HCOH	0.2135	0.4812	0.4846	0.4860	0.3232	0.3451	0.3434	0.3645	0.3082	0.3581	0.3717	0.3712
	MFH	0.1834	0.2399	0.2729	0.2731	0.3673	0.3752	0.3803	0.3815	0.3263	0.3374	0.3435	0.3451
	MVLH	0.1192	0.1347	0.1200	0.1202	0.3363	0.3339	0.3324	0.3284	0.3394	0.3401	0.3409	0.3499
	OMH-DQ	0.4177	0.6719	0.7414	0.7622	0.5223	0.5381	0.5823	0.5957	0.3949	0.4200	0.4446	0.4642
	SIDMH	0.6681	0.7479	0.7596	0.7660	0.5828	0.5976	0.6055	0.6120	0.4131	0.4277	0.4364	0.4706
	AMFH	**0.6837**	0.7501	0.7511	0.7519	0.6190	0.6240	0.6271	0.6385	0.4198	0.4374	0.4571	0.4887
	ASPQH	0.5328	**0.7723**	**0.7815**	**0.7712**	**0.6325**	**0.6511**	**0.6624**	**0.6556**	**0.4371**	**0.4620**	**0.4983**	**0.5128**



(a) Pascal Sentence (b) NUS-WIDE (c) IAPR TC-12

Fig. 3. The PR curves of all methods on Pascal Sentence (a), NUS-WIDE (b), and IAPR TC-12 (c).

Table 3
mAP comparison under different bits in unpaired scenarios.

Task	Methods	Pascal Sentence				NUS-WIDE				IAPR TC-12			
		16	32	64	128	16	32	64	128	16	32	64	128
I2I	ITQ	0.2559	0.2646	0.2751	0.2761	0.3164	0.3266	0.3357	0.3465	0.3501	0.3535	0.3542	0.3587
	LSH	0.0910	0.1076	0.1376	0.1718	0.3202	0.3265	0.3267	0.3269	0.3271	0.3379	0.3481	0.3504
	DLLH	0.2481	0.2599	0.2651	0.2706	0.3263	0.3264	0.3278	0.3368	0.3487	0.3503	0.3546	0.3555
	HCOH	0.1676	0.3326	0.3336	0.3360	0.3377	0.3406	0.3500	0.3647	0.3025	0.3135	0.3228	0.3356
	ASPQH_img	**0.3275**	**0.4648**	**0.6030**	**0.6150**	**0.3610**	**0.3677**	**0.4339**	**0.4360**	**0.3873**	**0.3920**	**0.3972**	**0.4096**
T2T	ITQ	0.4662	0.4879	0.4595	0.4386	0.3963	0.3896	0.3873	0.3856	0.3054	0.2976	0.2994	0.3012
	LSH	0.1803	0.2483	0.3551	0.3926	0.3283	0.3394	0.3507	0.3589	0.3109	0.3135	0.3219	0.3182
	DLLH	0.4512	0.4524	0.4726	0.4908	0.3739	0.3784	0.3803	0.3881	0.2963	0.2911	0.2891	0.2896
	HCOH	0.4859	0.6118	0.6226	0.6259	**0.6194**	0.6288	0.6316	0.6337	0.3856	0.4001	0.4164	0.4307
	ASPQH_txt	**0.5125**	**0.7519**	**0.7630**	**0.7623**	0.5894	**0.6303**	**0.6402**	**0.6409**	**0.3858**	**0.4090**	**0.4431**	**0.4497**

4.3. Retrieval accuracy comparison

In this section, we compare our proposed method with the baseline methods to validate the effectiveness of our model. In our experiments, we carry out three retrieval tasks including image query image, text query text, and image-text pair query image-text pair, which are often abbreviated to "I2I", "T2T" and "O2O", respectively. We report the retrieval performance of all methods under different code lengths on three datasets. The code length is set sequentially from 16 bits to 128 bits to evaluate the retrieval performance of our method comprehensively.

4.3.1. Results on Pascal sentence

The mAP values of all compared methods on the Pascal Sentence reported in Table 2. The proposed ASPQH outperforms all uni-modal hashing methods. Our proposed ASPQH is inferior to SIDMH and AMFH when the code length is 16 bits while ASPQH achieves an average improvement of 2.4% over the suboptimal baseline method (i.e., AMFH) when the code length is greater than 32 bits. In addition, we vary Hamming radius to return the points that fall within different radii and plot the Precision-Recall (PR) curves of all compared methods on the Pascal Sentence. As seen in Fig. 3(a), our proposed ASPQH performs best than all baselines. Further, we apply our model to uni-modal modality retrieval scenarios where only one single modality is provided and other modalities are missing. In our experiments, "ASPQH_img" indicates only the image modality is given in the encoding stage and "ASPQH_txt" denotes only the text modality is provided to complete hash encoding. It is noted that "ASPQH_img", and "ASPQH_txt" utilize the reconstructed features of the missing modality to obtain joint hash codes. In Table 3, we can see that our proposed ASPQH consistently outperforms all uni-modal methods by a large performance margin on the Pascal Sentence dataset. To observe the retrieval performance more intuitively, we provide the visualization of a testing experiment with an image-text query on the Pascal Sentence dataset. In Fig. 4, the left of the figure is a given query sample and the top five samples of the retrieved results of all methods are listed on the right of the figure. We can see that the first, third, and fifth positions of the shown results returned by our method are related to the given query sample. Compared with the baseline methods, our method has more competitive and achieves better performance.

Table 4
mAP comparison between SPQH and ASPQH under different bits.

Methods	Pascal Sentence				NUS-WIDE				IAPR TC-12			
	16	32	64	128	16	32	64	128	16	32	64	128
SPQH	0.5176	0.7641	0.7799	0.7689	0.6208	0.6231	0.6353	0.6410	0.4060	0.4601	0.4691	0.4964
ASPQH	**0.5328**	**0.7723**	**0.7815**	**0.7712**	**0.6325**	**0.6511**	**0.6624**	**0.6556**	**0.4371**	**0.4620**	**0.4983**	**0.5128**

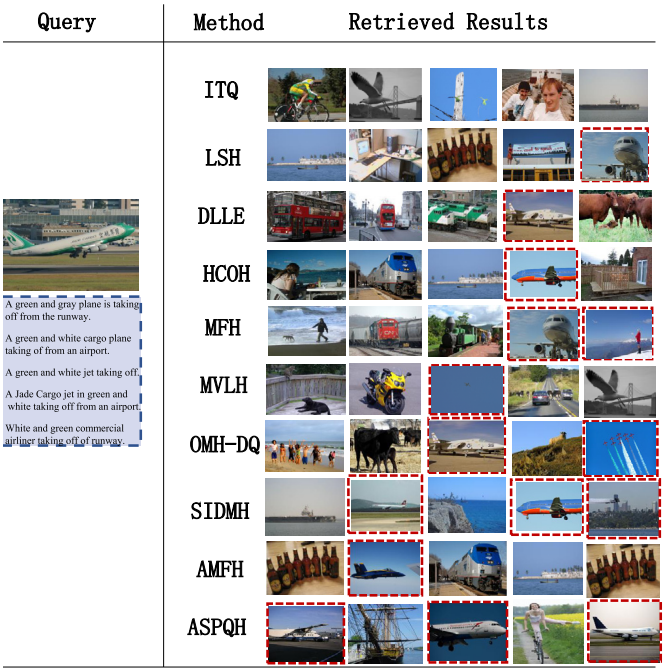


Fig. 4. Example of multi-modal retrieval. The images framed in red are the correct search results.

4.3.2. Results on NUS-WIDE

The mAP values of all baselines and our proposed method on the NUS-WIDE data set are shown in Table 2. We clearly find that ASPQH offers a significant improvement over the baseline methods. The PR curves of all methods on NUS-WIDE are plotted in Fig. 3 (b). By the above experiment results, we can see that our ASPQH is superior to all compared methods on the "O2O" task under different code lengths. In Table 3, we report the retrieval accuracy of all methods on "I2I" and "T2T" tasks. In terms of average accuracy, our proposed ASPQH is higher than the best baseline method (i.e., HCOH) by 5% on "I2I" task and achieves comparable performance on "T2T" task. The experimental results on NUS-WIDE show that our method can be used to handle large-scale image search and text search problems. Further, we can find that ASPQH_txt is better and closer to ASPQH than ASPQH_img by comparing ASPQH, ASPQH_img, and ASPQH_txt reported in Table 2, 3. These experimental phenomena indicate that our method is more able to handle unpaired textual queries than unpaired visual query scenarios on the NUS-WIDE dataset.

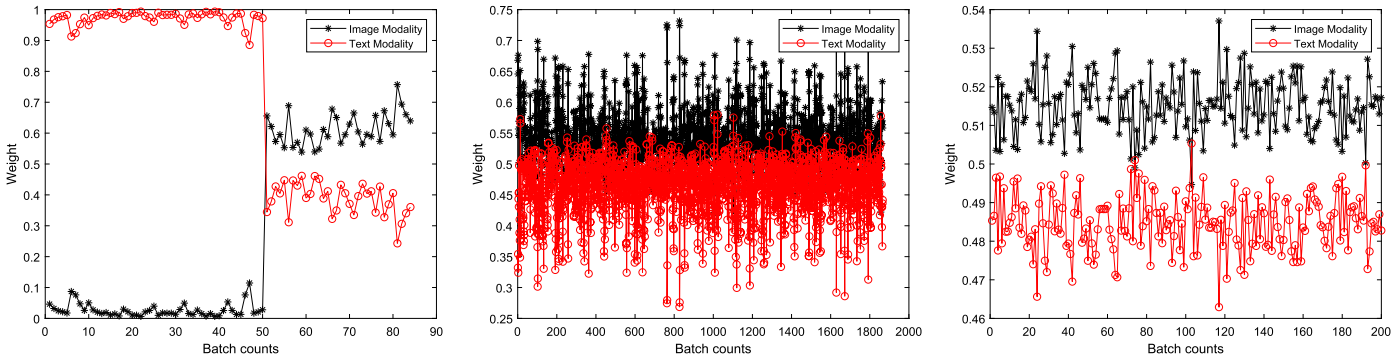
4.3.3. Results on IAPR TC-12

The mAP results and the PR curves of all methods on the "O2O" task are shown in Table 2 and Fig. 3, respectively. We can see that our proposed ASPQH consistently outperforms other methods and achieves higher accuracy with the increase of the code length. Specifically, our method achieves an average improvement of 3% on the "O2O" task. Likewise, our method in unpaired query scenarios is superior to other uni-modal hashing methods on the IARP TC-12 dataset. "ASPQH_img" and "ASPQH_txt" outperform the best baseline method by 4% and 1%, respectively.

4.4. Discussion

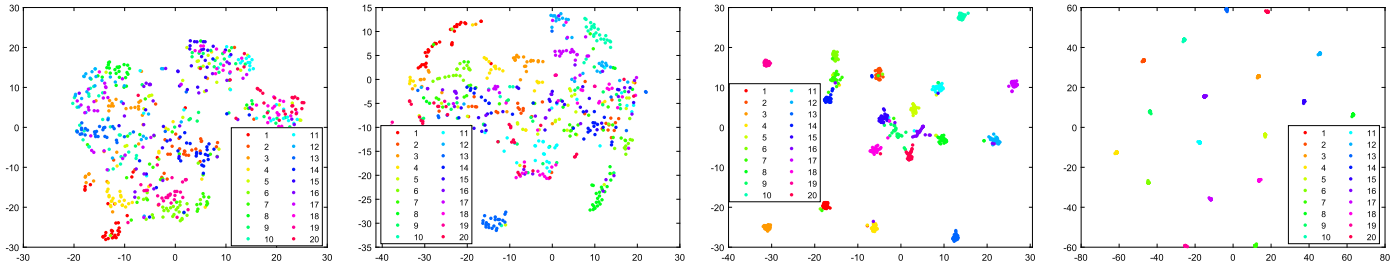
Most of the multi-modal hashing methods exploit mainly the complementary information between multi-modal data to learn joint hash codes. In this section, we experimentally explore the representation ability of different feature spaces during the space transformation process of the training stage. The experiments are performed on a single-label dataset (i.e., Pascal Sentence) to make an intuitive comparison. In experiments, we first select randomly 600 images and 600 texts to train our model and get the features of the intermediate process. Then, the t-SNE tool [19] is used to plot the feature distributions of these different spaces. Fig. 6 shows the visualization distributions of the original image features, the original text features, the features of common subspace, and binary features. As seen in Fig. 6 (a), (b), it is hard to separate them in the original multi-modal spaces. Taking text modality as an example, we show the distribution of common subspace in Fig. 6 (c). It can be observed that the different clusters are presented and the image-text pairs of different categories are gathered in different areas. Fig. 6 (d) presents the distribution of the learned joint hash codes. We can see clearly that the image-text pairs with the same class are more compact and the areas of different classes are more separated in Hamming space. The above experimental results demonstrate that our proposed ASPQH is very effective to learn discriminative hash codes.

In the encoding stage of our framework, we adopt the dynamic weighting module to eliminate the modality difference between the training set and query data. We design the following ablation experiments to investigate the effectiveness of our proposed ASPQH. "SPQH" indicates the weight of each modality is fixed as the learned weights in the training stage. We perform experiments to compare SPQH and ASPQH on three datasets. The experiment results under different code lengths are reported in Table 4. We can find that ASPQH is consistently better than SPQH. Specifically, ASPQH offers the average improvement of 0.68%, 2.1%, and 1.9% on Pascal Sentence, NUS-WIDE, and IAPR TC-12, respectively. It is noted that the improvements on NUS-WIDE and IAPR TC-12 are higher than that on Pascal Sentence. To visually observe the change of each modality weight in the encoding stage, we plot the variation curves of modality weights on each dataset. As seen in Fig. 5, the amplitude of curves on the NUS-WIDE and IAPR TC-12 dataset is large while that of the curves on Pascal Sentence is small. The possible reason is that the modality difference on NUS-WIDE and IAPR TC-12 is greater than that on Pascal Sentence. The above experimental results demonstrate that ASPQH can achieve more competitive performance than SPQH in the larger data difference scenarios. To explore the contribution of visual and textual features to the retrieval task, we conduct the ablation experiments in this part. ASPQH1 indicates that the proposed method only uses the visual feature to encode multimodal instance samples, which means that the weight of the visual modality is always set to 1 during the encoding phase, and the weight of the textual modality is set to 0. In ASPQH2, only the textual feature is utilized, that is, the weight of the visual modality is set to 0, and the weight of the textual modality is set to 1. As shown in Table 5, we can see that ASPQH1 has higher accuracy than ASPQH2 on three datasets. The retrieval results demonstrate that the visual feature is more important than textual feature to perform the multi-modal retrieval task on three datasets.



(a) Pascal Sentence (b) NUS-WIDE (c) IAPR TC-12

Fig. 5. The visualization of modality weights variation under different batches in the encoding stage.



(a) Original image features (b) Original text features (c) Low-dimensional subspace features (d) Binary features

Fig. 6. The t-SNE visualization of different feature spaces.

Table 5
mAP comparison based on different modality features.

Method	ASPQH1	ASPQH2	ASPQH
Pascal Sentence	0.6159	0.4837	0.7815
NUS-WIDE	0.4702	0.4323	0.6624
IAPR TC-12	0.4162	0.3680	0.4983

4.5. Parameter sensitivity analysis

In this section, we conduct experiments to observe the performance variation of our proposed method by varying one parameter value within an empirical candidate range and fixing the other parameters. In our experiments, the code length is set to 128 bits. The sensitivity analysis of λ, β, γ, and ρ is presented as follows.

λ is a balance parameter that affects the importance of different modalities. λ is tuned within the range of $[0.1, 0.9]$. The image modality has more effect on the performance of our model if its value is greater than 0.5, and vice versa. Fig. 7 (c) shows the performance variation under different λ values on three datasets. On Pascal Sentence and NUS-WIDE, the mAP value decreases when λ varies from 0.1 to 0.9. On the contrary, the mAP of our ASPQH is changed slightly on the IAPR TC-12 dataset with the increase of the λ. ASPQH achieves optimal retrieval performance when λ is set as 0.1, 0.1, and 0.9 on Pascal Sentence, NUS-WIDE, and IAPR TC-12, respectively. A possible reason is that the text modality plays a more important role than the image modality on Pascal Sentence and NUS-WIDE while the image modality contributes more information than the text modality on the IARP TC-12 dataset.

β controls the importance of the semantic preservation term of our model. In our experiments, β is determined in the range of $\{1e^{-5}, 1e^{-3}, 1e^{1}, 1e^{3}, 1e^{5}\}$. It can be found in Fig. 7 (a) that the mAP of our ASPQH will be lower if β is too small but the performance of our ASPQH is generally improved when β increases. Empirically, β is set as $1e^{1}$ on NUS-WIDE and the recommended range of β is $[1e^{3}, 1e^{5}]$ on Pascal Sentence and IAPR TC-12.

γ controls the over-fitting term of objective function (3). As seen in Fig. 7 (b), we can find that ASPQH achieves better performance when γ is not large. The mAP value of our model decreases quickly when γ is larger than $1e^{-3}$. Therefore, γ is recommended to choose from the range of $[1e^{-5}, 1e^{-3}]$.

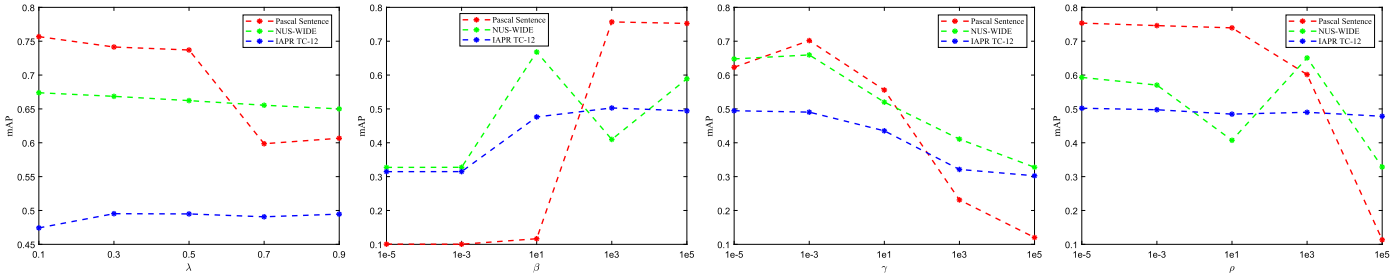
ρ is introduced to solve the discrete variable B of objective function (3). Fig. 7 (d) shows the experimental results on three datasets. We can observe that ASPQH can achieve stable performance with the change of ρ in the range of $[1e^{-5}, 1e^{5}]$ on the IAPR TC-12 dataset. Nevertheless, the performance deteriorates when its value is larger than $1e^{1}$ and $1e^{3}$ on Pascal Sentence and NUS-WIDE, respectively.

4.6. Convergence study

In this section, we explore experimentally the convergence of our proposed model. Fig. 8 shows the convergence curves of our method on three datasets when the hash code length is set to 128 bits. The shape of convergence curves of other code lengths is similar to that under the code length of 128 bits. In Fig. 8, the x-axis of each subfigure indicates the number of iterations and the y-axis represents the objective value. We can see that the objective function decreases quickly and rapidly converges to stable values on three datasets. It can be observed that our model converges within 30 iterations on three datasets. Especially on Pascal Sentence, the model converges within 10 iterations.

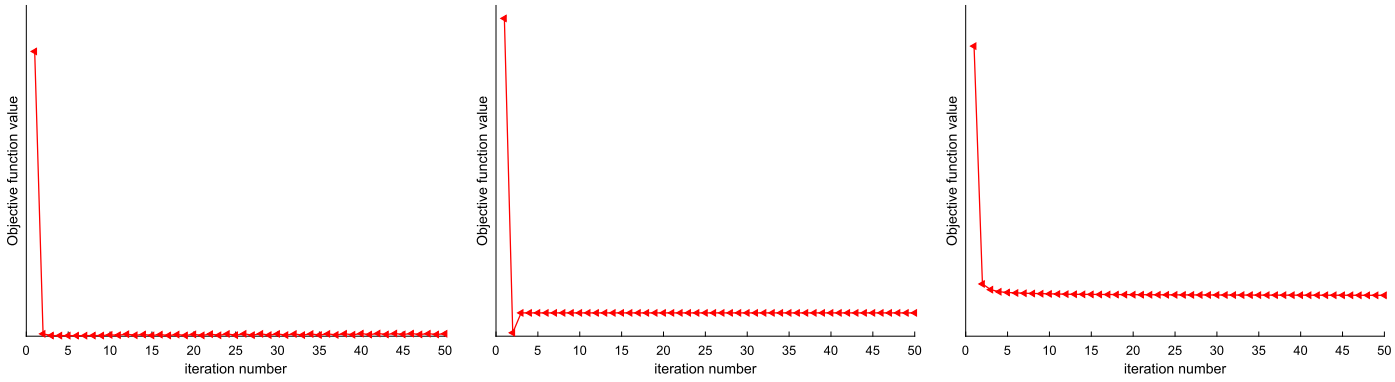
5. Conclusion

In this paper, we proposed an adaptive semi-paired query hashing method to fuse the discriminative semantic structure information and the complementary multi-modal information for multi-



(a) (b) (c) (d)

Fig. 7. The performance variations under different parameter settings on three datasets.



(a) Pascal Sentence (b) NUS-WIDE (c) IAPR TC-12

Fig. 8. The convergence curves of our proposed method on Pascal Sentence (a), NUS-WIDE (b), and IAPR TC-12 (c).

modal hash retrieval. A cross-modal reconstruction learning module is designed to enhance the semantic consistency between paired different modalities. In the encoding stage, we can obtain the joint hash codes of unpaired instances, based on the learned cross-modal reconstruction matrices. The self-weighting learning method is proposed to capture the modality difference between query data. Besides, we develop a fast optimization method to effectively solve our objective function. The experimental results demonstrate that our proposed method has superior retrieval performance and can be effectively applied to generate the joint hash codes for semi-paired multi-modal data.

The main limit of this research work is that the entire algorithm is designed based on manually extracted features. In future work, we will develop new fusion strategies and discrete optimization methods in a deep learning framework to further improve retrieval performance.

CRediT authorship contribution statement

Jun Yu: Methodology, Software, Validation, Writing – original draft, Writing – review & editing. **Wei Huang:** Methodology, Writing – review & editing. **Zuhe Li:** Methodology, Writing – review & editing. **Kunlin Li:** Conceptualization, Writing – review & editing. **Zhenqiu Shu:** Writing – review & editing. **Qi Wang:** Software, Visualization, Writing – review & editing. **Jinge Jiao:** Software, Visualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The paper is supported by the Research Startup Project of Zhengzhou University of Light Industry (No. 2021BSJJ025), the Henan Provincial Department of Science and Technology Research Project (Grant Nos. 222102210064, 222102210010, 212102210095), Academic Degrees & Graduate Education Reform Project of Henan Province (No. 2021SJGLX115Y).

References

[1] Z. An, T. Zhang, B. Ma, Y. Xu, Blind multicarrier waveform recognition based on spatial-temporal learning neural networks, Digit. Signal Process. 111 (2021) 102994.
[2] Z. An, T. Zhang, M. Shen, E. De Carvalho, B. Ma, C. Yi, T. Song, Series-constellation feature based blind modulation recognition for beyond 5 g mimo-ofdm systems with channel fading, IEEE Trans. Cogn. Commun. Netw. 8 (2022) 793–811.
[3] T.S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, 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, 2009, pp. 1–9.
[4] M. Datar, N. Immorlica, P. Indyk, V.S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the Twentieth Annual Symposium on Computational Geometry, 2004, pp. 253–262.
[5] H.J. Escalante, C.A. Hernández, J.A. Gonzalez, A. López-López, M. Montes, E.F. Morales, L.E. Sucar, L. Villasenor, M. Grubinger, The segmented and annotated iapr tc-12 benchmark, Comput. Vis. Image Underst. 114 (2010) 419–428.
[6] Y. Fang, Y. Ren, J.H. Park, Semantic-enhanced discrete matrix factorization hashing for heterogeneous modal matching, Knowl.-Based Syst. 192 (2020) 105381.
[7] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2012) 2916–2929.
[8] R. Ji, H. Liu, L. Cao, D. Liu, Y. Wu, F. Huang, Toward optimal manifold hashing via discrete locally linear embedding, IEEE Trans. Image Process. 26 (2017) 5411–5420.

[9] L. Jin, Z. Li, Y. Pan, J. Tang, Weakly-supervised image hashing through masked visual-semantic graph-based reasoning, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 916–924.
[10] G. Koutaki, K. Shirai, M. Ambai, Hadamard coding for supervised discrete hashing, IEEE Trans. Image Process. 27 (2018) 5378–5392.
[11] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing for scalable image search, in: 2009 IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 2130–2137.
[12] C.X. Li, T.K. Yan, X. Luo, L. Nie, X.S. Xu, Supervised robust discrete multimodal hashing for cross-media retrieval, IEEE Trans. Multimed. 21 (2019) 2863–2877.
[13] Z. Li, J. Tang, L. Zhang, J. Yang, Weakly-supervised semantic guided hashing for social image retrieval, Int. J. Comput. Vis. 128 (2020) 2265–2278.
[14] M. Lin, R. Ji, S. Chen, X. Sun, C.W. Lin, Similarity-preserving linkage hashing for online image retrieval, IEEE Trans. Image Process. 29 (2020) 5289–5300.
[15] M. Lin, R. Ji, H. Liu, Y. Wu, Supervised online hashing via Hadamard codebook learning, in: Proceedings of the 26th ACM International Conference on Multimedia, 2018, pp. 1635–1643.
[16] L. Liu, M. Yu, L. Shao, Multiview alignment hashing for efficient image search, IEEE Trans. Image Process. 24 (2015) 956–966.
[17] X. Lu, L. Liu, L. Nie, X. Chang, H. Zhang, Semantic-driven interpretable deep multi-modal hashing for large-scale multimedia retrieval, IEEE Trans. Multimed. 23 (2021) 4541–4554.
[18] X. Lu, L. Zhu, Z. Cheng, L. Nie, H. Zhang, Online multi-modal hashing with dynamic query-adaption, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 715–724.
[19] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[20] S. Pouyanfar, Y. Yang, S.C. Chen, M.L. Shyu, S. Iyengar, Multimedia big data analytics: a survey, ACM Comput. Surv. (CSUR) 51 (2018) 1–34.
[21] M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, Adv. Neural Inf. Process. Syst. 22 (2009).
[22] C. Rashtchian, P. Young, M. Hodosh, J. Hockenmaier, Collecting image annotations using Amazon's mechanical turk, in: Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, 2010, pp. 139–147.
[23] F. Shen, C. Shen, W. Liu, H.T. Shen, Supervised discrete hashing, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 37–45.
[24] X. Shen, F. Shen, L. Liu, Y.H. Yuan, W. Liu, Q.S. Sun, Multiview discrete hashing for scalable multimedia search, ACM Trans. Intell. Syst. Technol. (TIST) 9 (2018) 1–21.
[25] X. Shen, F. Shen, Q.S. Sun, Y.H. Yuan, Multi-view latent hashing for efficient multimedia search, in: Proceedings of the 23rd ACM International Conference on Multimedia, 2015, pp. 831–834.
[26] Z. Shu, Y. Bai, D. Zhang, J. Yu, Z. Yu, X.J. Wu, Specific class center guided deep hashing for cross-modal retrieval, Inf. Sci. 609 (2022) 304–318.
[27] J. Song, Y. Yang, Z. Huang, H.T. Shen, J. Luo, Effective multiple feature hashing for large-scale near-duplicate video retrieval, IEEE Trans. Multimed. 15 (2013) 1997–2008.
[28] T. Takahashi, T. Kurita, Mixture of subspaces image representation and compact coding for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 37 (2014) 1469–1479.
[29] D. Wang, X. Gao, X. Wang, L. He, Label consistent matrix factorization hashing for large-scale cross-modal similarity search, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2018) 2466–2479.
[30] D. Wang, Q. Wang, L. He, X. Gao, Y. Tian, Joint and individual matrix factorization hashing for large-scale cross-modal retrieval, Pattern Recognit. 107 (2020) 107479.
[31] Y. Wang, X. Luo, L. Nie, J. Song, W. Zhang, X.S. Xu, Batch: a scalable asymmetric discrete cross-modal hashing, IEEE Trans. Knowl. Data Eng. 33 (2020) 3507–3519.
[32] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, S. Yan, Cross-modal retrieval with cnn visual features: a new baseline, IEEE Trans. Cybern. 47 (2016) 449–460.
[33] J. Yu, W. Huang, Z. Li, Z. Shu, L. Zhu, Hadamard matrix guided multi-modal hashing for multi-modal retrieval, Digit. Signal Process. 130 (2022) 103743.
[34] J. Yu, X.J. Wu, J. Kittler, Semi-supervised hashing for semi-paired cross-view retrieval, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 958–963.
[35] J. Yu, X.J. Wu, D. Zhang, Unsupervised multi-modal hashing for cross-modal retrieval, Cogn. Comput. 14 (2022) 1159–1171.
[36] J. Yu, D. Zhang, Z. Shu, F. Chen, Adaptive multi-modal fusion hashing via Hadamard matrix, Appl. Intell. (2022) 1–15.
[37] D. Zhang, X.J. Wu, Scalable discrete matrix factorization and semantic autoencoder for cross-media retrieval, IEEE Trans. Cybern. 52 (2022) 5947–5959.

Jun Yu received his Ph.D. degree in Pattern Recognition from Jiangnan University and 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.

Wei Huang received the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2015. He is currently an Associate Professor with the School of Computer and Communication Engineering, Zhengzhou University of Light Industry. His current research interests include machine learning, pan-sharpening, hyperspectral image classification, hyperspectral image unmixing, remote sensing image detection.

Zuhe Li is an associate professor at the Zhengzhou University of Light Industry. He received his BS degree in electronic information science and technology from the Zhengzhou University of Light Industry in 2004, his MS degree in communication and information system from the Huazhong University of Science and Technology in 2008, and his Ph.D. degree in information and communication engineering from the Northwestern Polytechnical University in 2017. His current research interests include computer vision, sentiment analysis, and machine learning.

Kunlin Li received his Bachelor of Engineering degree from Anyang Institute of Technology in 2022. He is currently pursuing a Master of Engineering degree in the School of Computer Science and Communication Engineering at Zhengzhou University of Light Industry. His research interests include knowledge graph, data mining, and machine learning.

Zhenqiu Shu received his Ph.D. degree in computer applications from Nanjing University of Science and Technology in 2015. He is currently an associate professor in the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. His research interests include image processing, computer vision, and machine learning.

Qi Wang received the M.S. degree from School of Electrical Engineering, North China University of Water Resources and Electric Power, Zhengzhou, China, in 2017. She is currently ICT Manager in Zhengzhou Branch Company of China Unicom. Her research interests include computer vision and artificial intelligence.

Jinge Jiao received the M.S. degree from School of Electrical Engineering, North China University of Water Resources and Electric Power, Zhengzhou, China, in 2016. He is currently ICT Manager in Zhengzhou Branch Company of China Unicom. His research interests include computer vision and artificial intelligence.