Discrete Joint Semantic Alignment Hashing for Cross-Modal Image-Text Search

Song Wang, Huan Zhao, and Keqin Li, *Fellow, IEEE*

***Abstract*— Supervised cross-modal image-text hashing has aroused extensive concentrations in comprehending the correspondence between vision and language for data search tasks. Existing methods learn the compact hash codes by leveraging a given image-text data pairs or supervised information to explore such correspondence. However, they still confront obvious drawbacks. First, there is no engagement between multiple semantic information that yields the suboptimal search performance. Second, most of them adopt continuous relaxation strategy by discarding the discrete constraints, which results in large binary quantization errors. To deal with these problems, we propose a novel supervised hashing method, termed Discrete Joint Semantic Alignment Hashing (DJSAH). Specifically, it builds a connection between semantics (a.k.a. class labels and pairwise similarities) by the joint semantic alignment learning. And thus the high-level discriminative semantics can be preserved into the hash codes. Besides, a well-designed discrete optimization algorithm with linear computation and memory cost is developed to reduce the information loss of the hash codes with no need for relaxation. Extensive experiments and analyses on three benchmark datasets validate the superiority of the proposed DJSAH against several state-of-the-art hashing methods.**

***Index Terms*— Cross-modal image-text search, hash code, semantic alignment, supervised hashing.**

I. Introduction

CROSS-MODAL image-text search plays a crucial role in bridging the modality gap between vision and language understanding which aims to measure the semantic relevances between image-text data instances [1], [34], [46], [55]. Owing to the explosive roaming multimedia data on the Internet recently, hashing-based cross-modal image-text search methods that convert the original high-dimensional feature matrices into compact binary codes, have been a mainstream research topic in the multimedia information processing and computer vision community [2], [17], [33], [48]. Based on the advantage of the binary representation in high efficiency and low storage, a variety of researches have been proposed and obtained desirable progress for image-text search tasks [12], [27], [29], [51], [57], [60].

Due to the modality gap between image and text data, the most significant problem of cross-modal image-text hashing is how to map the original data instances into a shared binary hash space and meanwhile preserve the cross-modal semantic consistency. Specifically, existing hashing approaches can be generally grouped into two lines: (1) unsupervised ones [9], [37], [40], [44], [62], [65] aiming to explore the relevances by the original image and text data modalities, (2) supervised ones [21], [23], [39], [41], [54], [59] aiming to mine the correspondences by taking the shared supervised information (a.k.a. class labels or pairwise similarity matrix) of image-text data pairs into consideration. Generally speaking, supervised hashing could achieve more satisfactory performance than unsupervised hashing, which also becomes the focus of the following work.

A lot of supervised hashing methods [3], [4], [35], [42], [47], [50], [58] have been widely studied to promote the performance in the tasks of cross-modal image-text search. Such supervised methods are mainly designed by the large semantic similarity construction or the common label information to learn the unified hash codes for the search tasks. For instance, some representative works include Supervised Matrix Factorization Hashing (SMFH) [23], Semantic Correlation Maximization (SCM) [59], Label Consistent Matrix Factorization Hashing (LCMFH) [41] and Subspace Relation Learning for Cross-modal Hashing (SRLCH) [36], etc. More recently, benefiting from the renaissance in the techniques of deep learning, relevant researches (i.e., supervised deep hashing [14], [20], [61]) have been developed to tackle the problem of cross-modal image-text search. Although achieving the thrilling success, such methods are much time-consuming during training and cannot handle the proposed complex objective functions of the hashing models, making them unacceptable to the applications of large-scale cross-modal image-text data retrieval.

Despite the impressive search performance of supervised hashing methods so far, there are still several important problems worthy of consideration. (1) Limited semantic relations. Most pioneer methods yield the unsatisfactory search performance without fully exploiting the semantic relations. For example, LCMFH [41] and SRLCH [36] only utilize the class label information to generate the hash codes with no consideration for the pairwise similarity of inter-modality.

Manuscript received 11 August 2021; revised 30 January 2022; accepted 20 June 2022. Date of publication 27 June 2022; date of current version 28 October 2022. This work was supported by the National Natural Science Foundation of China under Grant 62076092 and Grant 61772188. This article was recommended by Associate Editor Z. He. *(Corresponding author: Huan Zhao.)*

Song Wang and Huan Zhao are with the College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China (e-mail: swang17@hnu.edu.cn; hzhao@hnu.edu.cn).

Keqin Li is with the College of Computer Science and Electronic Engineering, Hunan Uiversity, Hunan 410082, China, and also with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2022.3186714.

Digital Object Identifier 10.1109/TCSVT.2022.3186714

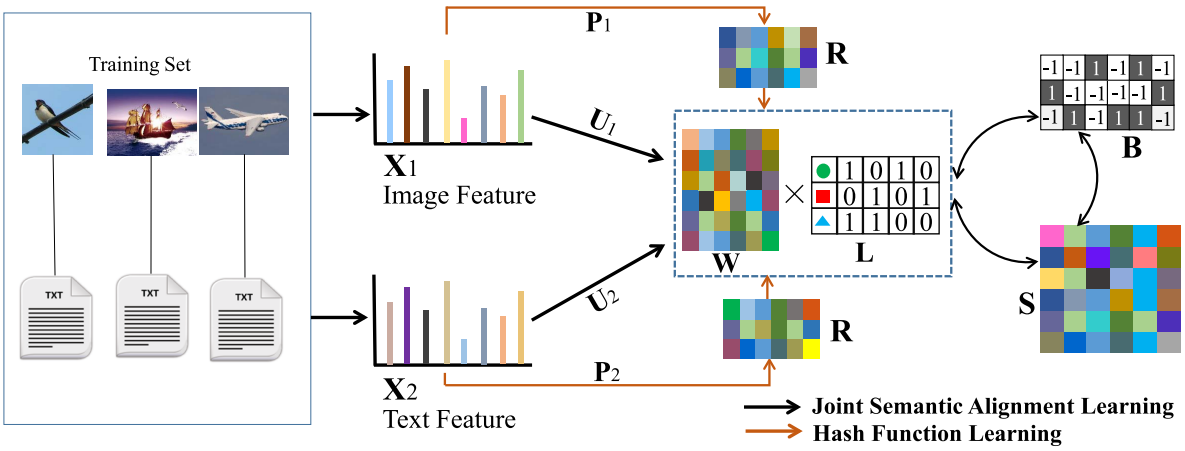


Fig. 1. The basic framework of the proposed DJSAH, which includes joint semantic alignment learning (black) and hash function learning (orange). Concretely, the proposed DJSAH generates the unified hash code $\mathbf{B}$ by integrating the original data distributions $\mathbf{X}_1, \mathbf{X}_2$ and the semantic alignment between the labels $\mathbf{L}$ and pairwise similarity matrix $\mathbf{S}$. Meanwhile, the semantic correspondences between distinct modalities are maintained into the to-be-learned hash codes. Besides, we adopt the linear regression to guide the modality-specific hash functions $\mathbf{P}_1, \mathbf{P}_2$.

SMFH [23], SCM [59], SePH [21] learn the hash codes for search tasks by the pairwise similarity matrix ignoring the shared label information, while consuming much computation and storage cost. The above two strategies cannot provide complementary semantic information for training, which lead to the large semantic information loss and thus achieve undesirable performance. (2) Binary optimization problem. As for hash optimization, many methods [21], [23], [36], [41], [59] adopt the continuous relaxation strategy by discarding the discrete constraints to produce approximate solutions of the hash codes. However, it inevitably brings up the large binary quantization errors and consequent less discriminative and inferior hash codes. Consequently, these methods yield the suboptimal search performance for the applications of cross-modal image-text datasets.

To tackle the aforementioned two problems, a novel supervised cross-modal image-text hashing method named Discrete Joint Semantic Alignment Hashing (DJSAH) is proposed in this work. Specifically, it incorporates the semantic consistency between the shared class labels and the transformed pairwise similarities as well as considering into account the original image-text data distributions to achieve the discriminative unified hash codes. Thus, the DJSAH method can full use of the multiple semantic relations rather than the limited supervision to enhance the search performance. In the meanwhile, we design a novel optimization algorithm instead of the traditional iterative optimization algorithm for solving the proposed complex objective function. And the proposed algorithm is devised to produce the discrete hash codes directly, thus alleviating the large binary quantization errors. The basic flowchart of DJSAH is illustrated in Figure 1. And our work possesses the following main contributions:

- We propose a novel supervised cross-modal image-text hashing method (DJSAH). It elaborately integrates the low-level data characteristics and the high-level multiple semantic consistency based the class labels and pairwise similarities to generate effective hash codes. In specific, this method not only upgrades the discriminative ability of the learned hash codes by fully leveraging semantic relations, but also circumvents considerable computation and storage cost by constructing the pairwise similarity.
- We develop a well-designed objective function and discrete optimization algorithm, which are able to efficiently obtain the discrete hash codes with no need for the relaxation. Moreover, the detailed theoretical analysis of the optimization algorithm is provided in this paper.
- Extensive experiments and analyses on three public datasets verify that our DJSAH can work better, especially the fully used multiple semantic relationships and discrete optimization algorithm, and thus achieving the state-of-the-art performance with regard to different evaluation metrics.

We arrange the rest of this paper. Section II highlights the related works of existing hashing approaches while Section III introduces the details of our method. Section IV presents experiments and analyses. Section V summarizes this paper.

II. Related Work

Hashing-based cross-modal image-text search methods have become a widespread research topic in the field of multi-modal information retrieval. In the following, we briefly introduce the related works which influence our study. They mainly consist of the two types below.

Unsupervised hashing methods realize the search tasks by exploring the semantic relevance of training instances based on the intrinsic data distributions. For example, Kumar and Udupa [18] maintained the similarity of identical samples by minimizing the weighted distance for cross-modal retrieval. Zhu *et al.* [65] adopted the anchor maps to keep the relationship of same instance instead of using the similarity maps. Zhou *et al.* [62] produced the unified hash codes by integrating sparse coding and matrix factorization to project the data instances into a binary space. Ding *et al.* [9] exploited the matrix factorization tool based cross-modal hashing to learns the hash codes. Wang *et al.* [43] leveraged the discrete collective matrix factorization and classification models to

construct the unified hashing framework. In short, the above methods mainly utilize the common data characteristics to train the hashing models. Different from the aforementioned researches, several works incorporate the common and unique data properties to improve the search accuracy in recent years. Wang *et al*. [44] obtained the high-quality hash codes by leveraging shared feature representation of the data instances while considering specific feature representations. Other relevant unsupervised studies are reported in [28], [38], [40]. Among methods only utilize the inner data distributions and characteristics of the original instances to obtain the hash codes. However, the individual feature representation of data pairs cannot fully explore the correlations across different modalities. Therefore, it is necessary to leverage the shared supervised information (labels or pairwise data constraints) in hashing learning.

Differently, supervised hashing methods further upgrade the quality of hash codes by leveraging the shared supervised information of data instances, which could work better than unsupervised ones. For example, a few methods [23], [39] learn the hash codes by preserving semantic relationship of the labels-similarity and data characteristics. Lin *et al*. [21] converted the affinity matrix into a probability distribution and close to it as the binary codes. However, such methods possess high space cost and heavy computational cost while training the hashing models due to the structure of the pairwise similarity matrix based on the supervised class information. To tackle this, a few hashing methods are well designed by employing the transformed supervised information and useful data characteristics to capture the discriminative unified hash codes. Thus, these works can contribute to improving the search accuracy as well as reducing the time complexity of the trained models. For instance, Zhang *et al*. [59] obtained the hash codes by closing to the similarity graphs based the label information. Jiang *et al*. [15] designed a discrete factor model to directly learn the hash codes when performing training. The authors well improve the accuracy as well as a fairly fast search speed for efficient cross-modal similarity search. Wang *et al*. [41] directly guided the hash function learning by the semantic label and matrix factorization. Shen *et al*. [36] combined the transformed semantic labels with subspace relation to jointly yield the discriminative hash codes. Nie *et al*. [31] embedded double semantic supervision information and a optimization strategy to enhance the search accuracy while keeping good learning efficiency. Wang *et al*. [45] learned a maximized hashing factor by jointly integrating multiple data distributions to perform image-text matching.

Very recently, there are several supervised methods [8], [16], [30], [32], [49] by adopting deep neural networks to generate the hash codes. For instance, Zhu *et al*. [64] incorporated the discriminative multi-view features and a discrete optimization method to obtain the hash codes. Cui *et al*. [6] designed the unified deep hash learning framework by jointly leveraging the hash functions and image representations for yielding the enhanced hash codes. Liu *et al*. [24] employed two image-text generation models and knowledge distillation framework to improve the retrieval performance of image-text matching. Generally, although delivering good retrieval results, these deep hashing approaches require heavy computational overhead and storage space for the training model, as well as the tuning and configuration of extensive super-parameter settings. Therefore, while seeking common progress in this field, it is quite important to complement each other and ignore their weaknesses. In hashing learning, this paper highlights the shallow supervised methods rather than deep methods for the direction of cross-modal image-text search.

III. Proposed Methodology

A. Notation and Problem Definition

Suppose that a cross-modal training set contains n image-text pairs, i.e., $\mathcal{O} = \{o_i\}_{i=1}^n$, $o_i = \left\{x_1^i, x_2^i, l_i\right\}$, where $x_1^i \in \mathbb{R}^{d_1}$, $x_2^i \in \mathbb{R}^{d_2}$ and $l_i \in \mathbb{R}^n$ denote the image, text and shared label feature vectors, respectively (usually $d_1 \neq d_2$). Let $\mathbf{X}_1 = \left\{x_1^i\right\}_{i=1}^n \in \mathbb{R}^{d_1 \times n}$, $\mathbf{X}_2 = \left\{x_2^i\right\}_{i=1}^n \in \mathbb{R}^{d_2 \times n}$ are the image and text feature matrices produced by corresponding vectors in $\mathcal{O}$. The supervision is the shared class label matrix $\mathbf{L} \in \mathbb{R}^{c \times n}$.

To facilitate the task of cross-modal image-text search, we construct a pairwise similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$. Following in [59], the similarity of o_i and o_j is defined as:

$$\tilde{\mathbf{S}}_{ij} = \frac{l_i \cdot l_j}{\|l_i\|_2 \left\|l_j\right\|_2}, \tag{1}$$

where l_i, l_j are two label vectors. $\|\cdot\|_2$ denotes L_2 norm. Utilizing the matrix $\tilde{\mathbf{L}} \in \mathbb{R}^{c \times n}$ to store the label information with $\tilde{\mathbf{L}}_{ki} = \frac{l_{ki}}{\|l_i\|_2}$, where $\tilde{\mathbf{L}}_{ki}$ is the element at the kth row and the ith column in $\tilde{\mathbf{L}}$, we have the similarity matrix $\tilde{\mathbf{S}} = \tilde{\mathbf{L}}^{\mathrm{T}}\tilde{\mathbf{L}}$. Then, the semantic pairwise similarity matrix $\mathbf{S} \in [-1, 1]^{n \times n}$ can be computed by:

$$\mathbf{S} = 2\tilde{\mathbf{S}} - \mathbf{E} = 2\tilde{\mathbf{L}}^{\mathrm{T}}\tilde{\mathbf{L}} - \mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}, \tag{2}$$

where $\mathbf{1}_n \in \{1\}^n$ and $\mathbf{E}$ denotes an all-one matrix.

It is commonly known that to match the data of different modalities accurately, they need to be in subspaces with similar feature structures. To fulfill this, it utilizes the kernelization [25], [63] to project the different data modalities into the common feature representations for acquiring the nonlinear structural characteristics of the original image-text data pairs. In general, it introduces the gaussian radial basis function (RBF) kernel to achieve this purpose. Given the t-th modality data features $\mathbf{X}_t$, we get the kernel features $\phi(\mathbf{K}^{(t)})$ as the model input, i.e., $\phi(\mathbf{K}^{(t)}) = \left\{\phi\left(\mathbf{k}_i^t\right)\right\}_{i=1}^n$, where $\mathbf{K}^{(t)} = \{\mathbf{X}_1, \mathbf{X}_2\}$. Thereinto, the adopted RBF kernel is employed by

$$\phi_i(k^{(t)}) = exp\left(-\frac{\left\|k_i{}^{(t)} - b_i{}^{(t)}\right\|_2^2}{2\varepsilon^2}\right), \tag{3}$$

where $\left\{k_j^{(t)}\right\}_{j=1}^m$ is m anchor points by random selection, and ε denotes the kernel width calculated by $\varepsilon = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|k_i - b_j\right\|_2$. $\left\|k_i^{(t)} - b_i^{(t)}\right\|_2^2$ represents the squared Euclidean distance between two feature vectors. In this paper, we set $m = 500$. In the following, $\|\cdot\|_F^2$, $tr(\cdot)$ are the Frobenius norm and the trace operation of the a matrix, respectively. DJSAH aims to get the final hash code matrix $\mathbf{B} \in \{-1, 1\}^{k \times n}$ to represent the cross-modal image-text data instances for the

TABLE I
THE MAIN SYMBOLS USED IN DJSAH

Notation	Definition
$\mathbf{X}_t$	Image or text feature matrix
$\mathbf{U}_t$	Basic matrix for exploring training data
$\mathbf{W}$	Projection matrix for correlation information
$\mathbf{P}_t$	Projection matrix for learning hash function
$\mathbf{R}$	Auxiliary matrix for transforming the labels
$\mathbf{L}$	Shared Label matrix from the data instances
$\mathbf{B}$	To-be-generated hash code matrix
$\mathbf{S}$	Semantic pairwise similarity matrix
$\phi(\bullet)$	Kernel function of feature matrices
d_1, d_2	Dimensions of image-text data pairs
n	Number of the training data
c	Number of the classes
k	hash code length

search tasks. For a clear presentation, Table I defines several significant symbols in this study, which are the frequently-used valuable matrices in the following.

B. Model Formulation

In the following, we illustrate the details of the proposed DJSAH method. It includes two main parts: joint semantic learning alignment part and hash function learning part. And specifically, to overcome the problem of the limited relations, we design a joint semantic alignment learning module by the multiple semantic consistency between the supervised class labels and pairwise similarities to formulate the discriminative unified hash codes.

1) Joint Semantic Alignment Learning: In the cross-modal supervised hashing learning, it is critical to embed the semantic relationships into the hash codes. In general, those data instances with similar topics have similar encoding representations for the hash codes. For some typical supervised hashing methods [26], [53], [56], [59], they retain the pairwise similarities that contained the available semantic relations into the generated hash codes by exploiting the commonly-used inner product, thus obtaining satisfactory performance. Thus, the objective function is formulated as:

$$\min_{\mathbf{B}} \left\| k\mathbf{S} - \mathbf{B}^{\mathrm{T}}\mathbf{B} \right\|_F^2, \quad s.t. \ \ \mathbf{B} \in \{-1, 1\}^{k\times n}. \tag{4}$$

Directly optimizing Eq. (4) is difficult because of the integer constraint attached to $\mathbf{B}$. A common way to solve the formulation of Eq. (4) is to relax the constraint $\mathbf{B} \in \{-1, 1\}^{k\times n}$ to $\mathbf{B} \in [-1, 1]^{k\times n}$. However, this strategy loses the available information of the hash code matrix and inevitably reduces the retrieval accuracy. What's more, the computation and memory cost of storing $\mathbf{S}$ is $O(n^2)$, which makes the hashing model much time-consuming. Last but not least, the aforementioned studies ignore their shared class label information that may cause the high-level information loss.

Motivated by the above challenges, we substitute one of hash code matrix $\mathbf{B}$ via the transformed real-valued semantic labels and join the low-level data distribution information. It is worth noting that such transformation has two advantages. 1) It improves the discrimination of the hash codes. 2) It obtains the linear computation and memory cost in training (described in the Subsection of complexity analysis). Accordingly, we directly generate the unified hash code $\mathbf{B}$ by integrating the multiple semantic information including the class labels and pairwise similarities, while taking into account the data characteristics. The objective function of joint semantic alignment learning is:

$$\begin{aligned} \min_{\mathbf{U}_t,\mathbf{W},\mathbf{B}} & \sum_{t=1}^{2} \left\| \phi(\mathbf{X}_t) - \mathbf{U}_t\mathbf{W}\mathbf{L} \right\|_F^2 + \lambda_1 \left\| k\mathbf{S} - \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L} \right\|_F^2 \\ & + \lambda_2 \left\| \mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L} \right\|_F^2 + \lambda_4 \sum_{t=1}^{2} \left(\left\| \mathbf{U}_t \right\|_F^2 + \left\| \mathbf{W} \right\|_F^2 \right), \\ s.t. \ \ & \mathbf{B} \in \{-1, 1\}^{k\times n}. \end{aligned} \tag{5}$$

As for Eq. (5), the first item represents collective matrix factorization [9], [41] for mapping different data modalities into the feature representations. The next two items are the joint semantic alignment term that depends on the multiple supervision information and its corresponding semantic consistency to generate the hash code matrix $\mathbf{B}$. The last item denotes the regularization for averting over-fitting of the hashing model.

For the third item ($\|\mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L}\|_F^2$), the matrix variable $\mathbf{W}\mathbf{L}$ denotes the common feature representation of the original instance. Many supervised methods usually learn the binary hash codes by adopting the transformation term

$$\left\| \mathbf{B} - \mathbf{W}\mathbf{L} \right\|_F^2, \quad s.t. \ \ \mathbf{B} \in \{-1, 1\}^{k\times n}, \tag{6}$$

to obtain the closed-form solution of hash code, i.e.,

$$\mathbf{B} = \mathrm{sgn}\left(\mathbf{W}\mathbf{L}\right). \tag{7}$$

Thereinto, $\mathrm{sgn}(\cdot)$ indicates a kind of symbolic function that simplifies complex problems. Directly employing the operation of Eq. (7) can learn the hash codes, but it also brings the quantization loss, and the quantization loss inevitably affects the search performance of the hashing algorithms. Motivated by [10], the authors propose a Iterative Quantization (ITQ) method to minimize the quantization loss of the to-be-generated hash codes and learning the efficient hash functions by an orthogonal transformation matrix $\mathbf{R}$

$$\left\| \mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L} \right\|_F^2, \quad s.t. \ \ \mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{I}, \tag{8}$$

where $\mathbf{I}$ is the identity matrix. Updating Eq. (8) by the iterative optimization strategy, we get the solution of the hash code

$$\mathbf{B} = \mathrm{sgn}\left(\mathbf{R}\mathbf{W}\mathbf{L}\right). \tag{9}$$

Meanwhile, several successful works [28], [52] based on the ITQ study have demonstrated the advantage of integrating the ITQ strategy [10] and the proposed models. Accordingly, we adopt the formulation of Eq. (8) to yield the unified hash codes. In the following, we will demonstrate that the matrix variable $\mathbf{R}\mathbf{W}\mathbf{L}$ not only contributes to minimizing the quantization loss of the hash codes, but it also is the local

optimal solution of Eq. (5), namely, Orthogonal Invariance Theorem [7], in supplementary material I.

Theorem 1 proves that the local optimal solution of direct optimization Eq. (5) is not necessarily the optimal solution of the problem. There exists an orthogonal transformation matrix $\mathbf{R}$ such that $\mathbf{RWL}$ is also a local optimal solution of Eq. (5). Therefore, by minimizing the quantization loss to learn an orthogonal matrix $\mathbf{R}$, the quantization loss of the hash codes can be reduced.

2) Hash Function Learning: In hash function learning, DJSAH designs to learn a series of hash functions by projecting image-text data into a shared Hamming space with the following formulation:

$$f\left(\phi\left(x_t\right)\right) = \operatorname{sgn}\left(\mathbf{P}_t\phi\left(x_t\right)\right), \tag{10}$$

where $\mathbf{P}_t$ denotes the projection matrix for the image and text modality. Among the above existing methods, the hash functions of distinct modalities are learned by a widely-used linear regression from feature representations to the hash codes. And the hash functions are linear and simple, which have the following formulation:

$$\min_{\mathbf{P}_t} \lambda_3 \left\|\mathbf{B} - \mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right\|_F^2 + \lambda_4\|\mathbf{P}_t\|_F^2. \tag{11}$$

To alleviate the information loss when optimizing the to-be-learned mapping matrix $\mathbf{P}_t$, we substitute the discrete $\mathbf{B}$ with the real-valued semantic label matrix $\mathbf{RWL}$. Meanwhile, employing Eq. (10) usually produces the quantization error and subsequent drop in search performance. Thus, Eq. (11) can be derived as:

$$\min_{\mathbf{W},\mathbf{P}_t} \lambda_3\sum_{t=1}^{2}\left\|\mathbf{RWL}-\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right\|_F^2+\lambda_4\sum_{t=1}^{2}\left(\|\mathbf{W}\|_F^2+\|\mathbf{P}_t\|_F^2\right). \tag{12}$$

After obtaining matrices $\mathbf{W}$, $\mathbf{R}$ in the part of joint semantic alignment learning, the hash function learns to obtain the mapping matrix $\mathbf{P}_t$ by using Eq. (12) and then is used in the querying process.

3) Overall Objective Function: The main idea of the proposed DJSAH is to enhance the search performance by exploring the multiple semantic relationship and corresponding consistency between distinct modalities. To tackle this, it needs to learn the unified hash code by the joint semantic alignment learning term and modality-specific hash functions through the hash function learning term with the applications to cross-modal image-text search. Thus, we formulate the overall objective function of DJSAH by integrating Eqs. (5) and (12) into a unified framework:

$$\min_{\mathbf{U}_t,\mathbf{W},\mathbf{P}_t,\mathbf{B},\mathbf{R}} J\left(\mathbf{U}_t,\mathbf{W},\mathbf{P}_t,\mathbf{B},\mathbf{R}\right) \quad s.t.\ \ \mathbf{B}\in\{-1,1\}^{k\times n}, \tag{13}$$

where the detailed formulation is

$$\begin{aligned} J = &\sum_{t=1}^{2}\left\|\phi\left(\mathbf{X}_t\right)-\mathbf{U}_t\mathbf{WL}\right\|_F^2+\lambda_1\left\|k\mathbf{S}-\mathbf{B}^{\mathrm{T}}\mathbf{RWL}\right\|_F^2 \\ &+\lambda_2\left\|\mathbf{B}-\mathbf{RWL}\right\|_F^2+\lambda_3\sum_{t=1}^{2}\left\|\mathbf{RWL}-\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right\|_F^2 \\ &+\lambda_4\sum_{t=1}^{2}\left(\|\mathbf{U}_t\|_F^2+\|\mathbf{W}\|_F^2+\|\mathbf{P}_t\|_F^2\right), \end{aligned} \tag{14}$$

where $\mathbf{U}_t \in \mathbb{R}^{d_t\times k}$ is basis matrix for the data instances. $\mathbf{W} \in \mathbb{R}^{k\times c}$ is auxiliary matrix. $\mathbf{P}_t \in \mathbb{R}^{k\times d_t}$ is projection matrix for learning the image and text hash functions. λ_1, λ_2, λ_3 and λ_4 are trade-off parameters. The first three terms defines the part of joint semantic alignment learning. The four term denotes the hash function learning part. The last term is the regularization for averting over-fitting.

C. Optimization Algorithm

In general, optimizing Eq. 13 is a discrete learning problem, which cannot be directly solved. Most pioneering methods utilize a continuous relaxation strategy that thresholds the relaxed solutions to the unified hash codes. However, when training the hashing models, this strategy inevitably leads to the loss of available information and subsequent inferior hash code. To alleviate this challenge, we design a discrete optimization algorithm to gain the matrix valuables of Eq. (13) in this paper. The optimization problem can be handled iteratively in the following description.

1. **Updating $\mathbf{U}_t$**. By fixing $\mathbf{P}_t$, $\mathbf{W}$, $\mathbf{R}$ and $\mathbf{B}$, Eq. (13) can be derived as:

$$\min_{\mathbf{U}_t}\sum_{t=1}^{2}\left\|\phi\left(\mathbf{X}_t\right)-\mathbf{U}_t\mathbf{WL}\right\|_F^2+\lambda_4\sum_{t=1}^{2}\left\|\mathbf{U}_t\right\|_F^2. \tag{15}$$

Setting the derivative of Eq. (15) with regard to $\mathbf{U}_t$ equal as zero, we obtain

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{U}_t} &= -2\sum_{t=1}^{2}\left(\phi\left(\mathbf{X}_t\right)-\mathbf{U}_t\mathbf{WL}\right)\left(\mathbf{LW}\right)^{\mathrm{T}}+2\lambda_4\sum_{t=1}^{2}\mathbf{U}_t \\ &= -2\sum_{t=1}^{2}\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}+2\sum_{t=1}^{2}\mathbf{U}_t\left(\mathbf{WLL}^{\mathrm{T}}\mathbf{W}^{\mathbf{T}}+\lambda_4\mathbf{I}\right) \\ &= 0. \end{aligned} \tag{16}$$

Thus, the closed-form solution of $\mathbf{U}_t$ is:

$$\mathbf{U}_t = \sum_{t=1}^{2}\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\left(\mathbf{WLL}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}+\lambda_4\mathbf{I}\right)^{-1}. \tag{17}$$

2. **Updating $\mathbf{W}$**. By fixing other variable matrices, Eq. (13) is reduced as:

$$\begin{aligned} \min_{\mathbf{W}} &\sum_{t=1}^{2}\left\|\phi\left(\mathbf{X}_t\right)-\mathbf{U}_t\mathbf{WL}\right\|_F^2+\lambda_1\left\|k\mathbf{S}-\mathbf{B}^{\mathrm{T}}\mathbf{RWL}\right\|_F^2 \\ &+\lambda_2\|\mathbf{B}-\mathbf{RWL}\|_F^2+\lambda_3\sum_{t=1}^{2}\|\mathbf{RWL}-\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\|_F^2+\lambda_4\|\mathbf{W}\|_F^2. \end{aligned} \tag{18}$$

Setting the derivative of Eq. (18) with regard to $\mathbf{W}$ equal as zero, we have

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{W}} = &-2\sum_{t=1}^{2}\mathbf{U}_t{}^{\mathrm{T}}\left(\phi\left(\mathbf{X}_t\right)-\mathbf{U}_t\mathbf{WL}\right)\mathbf{L}^{\mathrm{T}} \\ &-2\lambda_1\mathbf{R}^{\mathrm{T}}\mathbf{B}\left(k\mathbf{S}-\mathbf{B}^{\mathrm{T}}\mathbf{RWL}\right)\mathbf{L}^{\mathrm{T}} \\ &-2\lambda_2\mathbf{R}^{\mathrm{T}}\left(\mathbf{B}-\mathbf{RWL}\right)\mathbf{L}^{\mathrm{T}} \\ &+2\lambda_3\mathbf{R}^{\mathrm{T}}\sum_{t=1}^{2}\left(\mathbf{RWL}-\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right)\mathbf{L}^{\mathrm{T}}+2\lambda_4\mathbf{W} \end{aligned}$$

$$
\begin{aligned}
&= -2\sum_{t=1}^{2}\Big(\mathbf{U}_t{}^{\mathrm{T}}\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}} + \lambda_1 k\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}} + \lambda_2\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{L}^{\mathrm{T}} \\
&\quad + \lambda_3\mathbf{R}^{\mathrm{T}}\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\Big) \\
&\quad + 2\sum_{t=1}^{2}\Big(\mathbf{U}_t{}^{\mathrm{T}}\mathbf{U}_t + \lambda_1\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{R} \\
&\quad + \left(\lambda_2+\lambda_3\right)\mathbf{R}^{\mathrm{T}}\mathbf{R} + \lambda_4\mathbf{I}\Big)\mathbf{W}\mathbf{L}\mathbf{L}^{\mathrm{T}} \\
&= 0. \qquad (19)
\end{aligned}
$$

Thus, the closed-form solution of $\mathbf{W}$ is:

$$
\begin{aligned}
\mathbf{W} = &\sum_{t=1}^{2}\Big(\mathbf{U}_t{}^{\mathrm{T}}\mathbf{U}_t + \lambda_1\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{R} + \left(\lambda_2+\lambda_3\right)\mathbf{R}^{\mathrm{T}}\mathbf{R} + \lambda_4\mathbf{I}\Big)^{-1} \\
&\times \sum_{t=1}^{2}\Big(\mathbf{U}_t{}^{\mathrm{T}}\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}} + \lambda_1 k\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}} + \lambda_2\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{L}^{\mathrm{T}} \\
&+ \lambda_3\mathbf{R}^{\mathrm{T}}\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\Big)\left(\mathbf{L}\mathbf{L}^{\mathrm{T}}\right)^{-1}. \qquad (20)
\end{aligned}
$$

However, the calculation of $\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}}$ consumes $O(n^2)$, which produces high computation and memory efficiency. To tackle this, we substitute $\mathbf{S}$ by using Eq. (2), and then we have $\mathbf{R}^{\mathrm{T}}\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}} = 2\left(\mathbf{B}\tilde{\mathbf{L}}^{\mathrm{T}}\right)\left(\mathbf{R}\mathbf{L}\tilde{\mathbf{L}}^{\mathrm{T}}\right) - \left(\mathbf{B}\mathbf{1}_n\right)\left(\mathbf{R}\mathbf{L}\mathbf{1}_n\right)^{\mathrm{T}}$ which takes $O(n)$. Thus, the time complexity of updating $\mathbf{W}$ is $O(n)$.

3. **Updating $\mathbf{P}_t$**. By fixing the others, Eq. (13) with $\mathbf{P}_t$ as the only argument is derived:

$$
\min_{\mathbf{P}_t}\lambda_3\sum_{t=1}^{2}\left\|\mathbf{R}\mathbf{W}\mathbf{L} - \mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right\|_F^2 + \lambda_4\sum_{t=1}^{2}\left\|\mathbf{P}_t\right\|_F^2. \qquad (21)
$$

We set the derivative of Eq. (21) with regard to $\mathbf{P}_t$ equal to zero, and then we derive that

$$
\begin{aligned}
\frac{\partial J}{\partial \mathbf{P}_t} &= -2\lambda_3\sum_{t=1}^{2}\left(\mathbf{R}\mathbf{W}\mathbf{L} - \mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right)\phi(\mathbf{X}_t)^{\mathrm{T}} + 2\lambda_4\sum_{t=1}^{2}\mathbf{P}_t \\
&= -2\lambda_3\sum_{t=1}^{2}\mathbf{R}\mathbf{W}\mathbf{L}\phi(\mathbf{X}_t)^{\mathrm{T}} + 2\sum_{t=1}^{2}\mathbf{P}_t\left(\phi\left(\mathbf{X}_t\right)\phi(\mathbf{X}_t)^{\mathrm{T}} + \lambda_4\mathbf{I}\right) \\
&= 0. \qquad (22)
\end{aligned}
$$

Thus, the closed-form solution of $\mathbf{P}_t$ is:

$$
\mathbf{P}_t = \lambda_3\sum_{t=1}^{2}\mathbf{R}\mathbf{W}\mathbf{L}\phi(\mathbf{X}_t)^{\mathrm{T}}\left(\lambda_3\phi\left(\mathbf{X}_t\right)\phi(\mathbf{X}_t)^{\mathrm{T}} + \lambda_4\mathbf{I}\right)^{-1}. \qquad (23)
$$

4. **Updating $\mathbf{R}$**. By fixing the others, Eq. (13) with $\mathbf{R}$ as the only argument is reduced:

$$
\begin{aligned}
\min_{\mathbf{R}}\lambda_1\left\|k\mathbf{S} - \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right\|_F^2 + \lambda_2\left\|\mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L}\right\|_F^2 \\
+ \lambda_3\sum_{t=1}^{2}\left\|\mathbf{R}\mathbf{W}\mathbf{L} - \mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right\|_F^2. \qquad (24)
\end{aligned}
$$

Based on the identity between Frobenius matrix and trace matrix (e.g., $\|\mathbf{Q}\|_F^2 = tr\left(\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\right)$), Eq. (24) is rewritten as:

$$
\begin{aligned}
&\min_{\mathbf{R}}\lambda_1 tr\left(-2k\mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\mathbf{S}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}(\mathbf{W}\mathbf{L})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{B}\right) \\
&\quad + \lambda_2 tr\left(-2\mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L} + (\mathbf{W}\mathbf{L})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right) \\
&\quad + \lambda_3 tr\left(-2\sum_{t=1}^{2}\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\mathbf{R} + \mathbf{R}\mathbf{W}\mathbf{L}\mathbf{W}\mathbf{L}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\right) + const, \qquad (25)
\end{aligned}
$$

where *const* represents a constant item $tr\left(\lambda_1 k^2\mathbf{S}\mathbf{S}^{\mathrm{T}} + \lambda_2\mathbf{B}^{\mathrm{T}}\mathbf{B} + \lambda_3\mathbf{P}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{P}^{\mathrm{T}}\right)$ that is not related to matrix $\mathbf{R}$. According to the derivatives formula of trace matrix, Eq. (25) can be further derived. Letting the partial derivative of Eq. (25) concerning $\mathbf{R}$ equal 0, we have

$$
\begin{aligned}
&-\lambda_1\left(2k\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} - 2\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{R}\left(\mathbf{W}\mathbf{L}\right)\left(\mathbf{W}\mathbf{L}\right)^{\mathrm{T}}\right) \\
&-\lambda_2\left(2\mathbf{B}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} - 2\mathbf{R}\left(\mathbf{W}\mathbf{L}\right)\left(\mathbf{W}\mathbf{L}\right)^{\mathrm{T}}\right) \\
&-\lambda_3\sum_{t=1}^{2}\left(2\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} - 2\mathbf{R}\left(\mathbf{W}\mathbf{L}\right)\left(\mathbf{W}\mathbf{L}\right)^{\mathrm{T}}\right) \\
&\quad = 0 \qquad (26)
\end{aligned}
$$

With this transformation, we can easily get the closed-form solution of $\mathbf{R}$:

$$
\begin{aligned}
\mathbf{R} = &\left[\lambda_1\mathbf{B}\mathbf{B}^{\mathrm{T}} + \left(\lambda_2+\lambda_3\right)\mathbf{I}\right]^{-1}\sum_{t=1}^{2}\left[\left(\lambda_1 k\mathbf{B}\mathbf{S} + \lambda_2\mathbf{B}\right.\right. \\
&\left.\left. + \lambda_3\mathbf{P}_t\phi\left(\mathbf{X}_t\right)\right)\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\right]\left[\mathbf{W}\mathbf{L}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}\right]^{-1}, \qquad (27)
\end{aligned}
$$

which $\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}$ consumes $O(n^2)$. Analogous to $\mathbf{W}$, we obtain $\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}} = 2\left(\mathbf{B}\tilde{\mathbf{L}}^{\mathrm{T}}\right)\left(\mathbf{W}\mathbf{L}\tilde{\mathbf{L}}^{\mathrm{T}}\right)^{\mathrm{T}} - \left(\mathbf{B}\mathbf{1}_n\right)\left(\mathbf{W}\mathbf{L}\mathbf{1}_n\right)^{\mathrm{T}}$ by Eq. (2). Thus, the time complexity of $\mathbf{B}\mathbf{S}\mathbf{L}^{\mathrm{T}}\mathbf{W}^{\mathrm{T}}$ is reduced to $O(n)$.

5. **Updating $\mathbf{B}$**. With other variables fixed, we can get:

$$
\min_{\mathbf{B}\in\{-1,1\}^{k\times n}}\lambda_1\left\|k\mathbf{S} - \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right\|_F^2 + \lambda_2\left\|\mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L}\right\|_F^2. \qquad (28)
$$

Analogous to Eq. (25), Eq. (28) is equivalent to

$$
\begin{aligned}
\min_{\mathbf{B}}\lambda_1 tr\left(-2k\mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\mathbf{S}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}(\mathbf{W}\mathbf{L})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{B}\right) \\
+ \lambda_2 tr\left(-2\mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right) + const, \qquad (29)
\end{aligned}
$$

where *const* represents a constant item $tr\left(\lambda_1 k^2\mathbf{S}\mathbf{S}^{\mathrm{T}} + \lambda_2\mathbf{B}^{\mathrm{T}}\mathbf{B} + \lambda_2(\mathbf{W}\mathbf{L})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right)$ that is independent of the matrix variable $\mathbf{B}$. Specifically, $\mathbf{B}^{\mathrm{T}}\mathbf{B}$ is equal to a constant in terms of $\mathbf{B}\in\{-1,1\}^{k\times n}$. However, optimizing directly $\mathbf{B}$ is a NP-hard problem because of several discrete constraints on the hash codes [11], [19]. Accordingly, many researches [21], [23], [41], [59] adopt the continuous relaxation strategy, i.e., discarding or relaxing the irrelevant items to $\mathbf{B}$, which could result in large binary quantization loss. To tackle this, we design a novel discrete optimization algorithm by building on the augmented Lagrangian multiplier (ALM) [22], [26] strategy. Concretely, we introduce an auxiliary variable matrix $\mathbf{M}\in\{-1,1\}^{k\times n}$ to substitute one of the hash code matrix $\mathbf{B}$. Then Eq. (29) can be transformed as:

$$
\min_{\mathbf{B}}\lambda_1 tr\left(-2k\mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\mathbf{S}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}(\mathbf{W}\mathbf{L})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{M}\right)
$$

$$+\lambda_2 tr\left(-2\mathbf{B}^{\mathrm{T}}\mathbf{RWL}\right)+\frac{\theta}{2}\left\|\mathbf{B}-\mathbf{M}+\frac{\mathbf{N}}{\theta}\right\|_F^2, \quad (30)$$

where $\mathbf{N}\in\{-1,1\}^{k\times n}$ measures the differences between $\mathbf{B}$ and $\mathbf{M}$. Thereinto, the last item of Eq. (30) can be easily transformed as:

$$\begin{aligned}&\min_{\mathbf{B}}\frac{\theta}{2}\left\|\mathbf{B}-\mathbf{M}+\frac{\mathbf{N}}{\theta}\right\|_F^2\\&=\min_{\mathbf{B}}\frac{\theta}{2}tr\left(\left(\left(\mathbf{B}^{\mathrm{T}}-\left(\mathbf{M}-\frac{\mathbf{N}}{\theta}\right)^{\mathrm{T}}\right)\left(\mathbf{B}-\left(\mathbf{M}-\frac{\mathbf{N}}{\theta}\right)\right)\right)\right)\\&=\min_{\mathbf{B}}\frac{\theta}{2}tr\left(-2\mathbf{B}^{\mathrm{T}}\left(\mathbf{M}-\frac{\mathbf{N}}{\theta}\right)\right)\\&=\min_{\mathbf{B}}tr\left(-\theta\mathbf{B}^{\mathrm{T}}\mathbf{M}+\mathbf{B}^{\mathrm{T}}\mathbf{N}\right).\end{aligned} \quad (31)$$

Next, we remove the constant term of Eq. (31) unrelated to $\mathbf{B}$. Under the condition of $\mathbf{B}\in\{-1,1\}^{k\times n}$, Eq. (30) is further reduced as:

$$\begin{aligned}&\min_{\mathbf{B}}\lambda_1 tr\left(-2k\mathbf{B}^{\mathrm{T}}\mathbf{RWLS}^{\mathrm{T}}+\mathbf{B}^{\mathrm{T}}\mathbf{RWL}(\mathbf{WL})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{M}\right)\\&\qquad+\lambda_2 tr\left(-2\mathbf{B}^{\mathrm{T}}\mathbf{RWL}\right)+tr\left(-\theta\mathbf{B}^{\mathrm{T}}\mathbf{M}+\mathbf{B}^{\mathrm{T}}\mathbf{N}\right).\end{aligned} \quad (32)$$

Similar to the transformation form of Eq. (26), we can further simply Eq. (32) and easily get the final closed-form solution of $\mathbf{B}$:

$$\begin{aligned}\mathbf{B}=\operatorname{sgn}\Big(&2\lambda_1 k\mathbf{RWLS}^{\mathrm{T}}+2\lambda_2\mathbf{RWL}\\&-\lambda_1\mathbf{RWL}(\mathbf{WL})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{M}+\theta\mathbf{M}-\mathbf{N}\Big).\end{aligned} \quad (33)$$

As $\mathbf{R}$ does, we get $\mathbf{RWLS}=2\mathbf{RWL}\tilde{\mathbf{L}}^{\mathrm{T}}\tilde{\mathbf{L}}-\mathbf{RWL}\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}$. The time complexity of this item is reduced as $O(n)$.

6. **Updating M**. By fixing other variables, Eq. (30) is reduced as:

$$\begin{aligned}&\min_{\mathbf{M}}\lambda_1 tr\left(\mathbf{B}^{\mathrm{T}}\mathbf{RWL}(\mathbf{WL})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{M}\right)+\frac{\theta}{2}\left\|\mathbf{B}-\mathbf{M}+\frac{\mathbf{N}}{\theta}\right\|_F^2\\&\quad=\min_{\mathbf{M}}tr\left(\mathbf{M}^{\mathrm{T}}\left(\lambda_1\mathbf{RWL}(\mathbf{WL})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{B}-\theta\mathbf{B}-\mathbf{N}\right)\right).\end{aligned} \quad (34)$$

Similar to the optimization of $\mathbf{B}$, the solution of $\mathbf{M}$ is:

$$\mathbf{M}=\operatorname{sgn}\left(-\lambda_1\mathbf{RWL}(\mathbf{WL})^{\mathrm{T}}\mathbf{R}^{\mathrm{T}}\mathbf{B}+\theta\mathbf{B}+\mathbf{N}\right). \quad (35)$$

7. **Updating N**. By fixing other variables, we have the solution:

$$\mathbf{N}=\mathbf{N}+\theta\left(\mathbf{B}-\mathbf{M}\right). \quad (36)$$

Depending on the mathematical expressions of Eqs. (33), (35), (36), we conclude that the closed-form solution of the hash codes $\mathbf{B}$ is discretely obtained by the optimization algorithm during training. This avoids the large quantization loss reflected by the continuous relaxation strategy. Besides, Algorithm 1 reviews the optimization process of the proposed DJSAH method. The corresponding time complexity is mainly affected via the iteration number w in lines 4−12.

To clear show the interrelation between several to-be-trained matrix variables when performing optimization, we implement the network structure of the variables in the objective function. As in shown in Fig. 2, it is straightforward that we can easily know the connection relationship of the variables $\mathbf{U}_t$, $\mathbf{W}$, $\mathbf{P}_t$, $\mathbf{R}$, $\mathbf{M}$, $\mathbf{N}$. Meanwhile, we also directly find the relationship between to-be-trained variables and $\mathbf{X}_t$, $\mathbf{L}$, $\mathbf{S}$.

Algorithm 1 The DJSAH Algorithm

Input: Feature matrices $\mathbf{X}_t$, label matrix $\mathbf{L}$, parameters λ_1, λ_2, λ_3, λ_4, θ, iteration number w, binary code length k.
Output: Code matrix $\mathbf{B}$ and the projection matrix $\mathbf{P}_t$.

1: Randomly initialize $\mathbf{U}_t$, $\mathbf{W}$, $\mathbf{B}$, $\mathbf{P}_t$, $\mathbf{R}$.
2: Construct the pairwise similarity matrix $\mathbf{S}$ by the label matrix $\mathbf{L}$.
3: Randomly pick up the anchors to transform $\mathbf{X}_t$ as nonlinear features $\phi(\mathbf{X}_t)$.
4: **for** $j=1$ to w **do**
5: Calculate $\mathbf{U}_t$ using Eq. (17).
6: Calculate $\mathbf{W}$ using Eq. (20).
7: Calculate $\mathbf{P}_t$ using Eq. (23).
8: Calculate $\mathbf{R}$ using Eq. (27).
9: Calculate $\mathbf{B}$ using Eq. (33).
10: Calculate $\mathbf{M}$ using Eq. (35).
11: Calculate $\mathbf{N}$ using Eq. (36).
12: **end for**

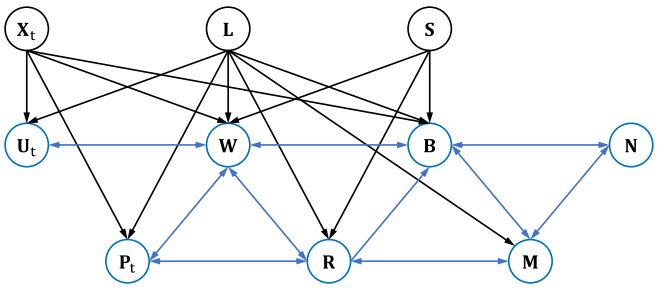


Fig. 2. The network structure of matrix variables during training, where the black hollow circles are the known variables and blue hollow circles are the to-be-trained variables.

D. Complexity Analysis

We analyzes the time complexity and space complexity of the proposed DJSAH in detail. For each iteration, the time complexity of our DJSAH contains $O\left(\left(k^2+kd_t+kc\right)n+c^2+k^3\right)$ for Eq. (17), $O\left(\left(k^2+kd_t+kc+k+c^2\right)n+d_tk^2+k^2+k^3+c^2+c^3\right)$ for Eq. (20), $O\left(\left(d_t^2+kd_t+kc\right)n+d_t^2+k^2+k^2c\right)$ for solving Eq. (23), $O\left(\left(k^2+kd_t+kc+k\right)n+c^2+k^2+k^3\right)$ for optimizing Eq. (27), updating $\mathbf{B}$ as $O(kcn)$ for Eq. (33), $O\left(\left(k^2+kc\right)n+ck^2+k^2+k^3\right)$ for Eq. (35) and $O(kn)$ for Eq. (36), which $d_t=\{d_1,d_2\}$ is the feature dimensions of the modalities and c,k,n,w are the number of classes, the hash code length, the number of training instances and the iteration number, respectively. Since $c,d,k\ll n$, the overall time complexity is $O(wn)$. Thus, the time complexity of DJSAH is linearly related to the size of training set n. As for the space complexity, we adopt the expression of Eq. (2) to substitute the calculation of $\mathbf{S}$ for constructing the linear memory cost, which reduces the memory cost from $O(n^2)$

to $O(n)$. As a result, the time and space complexity of the proposed method is linear to n.

E. Multi-Modalities Extension

Cross-modal image-text search is a fundamental but significant research topic in the field of multi-modal information retrieval. Although only concern image and text modalities in this work, our proposed DJSAH could be simply extended to multiple modalities. The multi-modal formulation of Eq. (37) is derived from the above bimodal Eq. (13):

$$\begin{aligned}\min_{\mathbf{U}_r,\mathbf{W},\mathbf{P}_r,\mathbf{B},\mathbf{R}} &\sum_r \|\phi(\mathbf{X}_r) - \mathbf{U}_r\mathbf{W}\mathbf{L}\|_F^2 + \lambda_1 \left\|k\mathbf{S} - \mathbf{B}^{\mathrm{T}}\mathbf{R}\mathbf{W}\mathbf{L}\right\|_F^2 \\ &+ \lambda_2 \|\mathbf{B} - \mathbf{R}\mathbf{W}\mathbf{L}\|_F^2 + \lambda_3 \sum_r \|\mathbf{R}\mathbf{W}\mathbf{L} - \mathbf{P}_r\phi(\mathbf{X}_r)\|_F^2 \\ &+ \lambda_4 \sum_r \left(\|\mathbf{U}_r\|_F^2 + \|\mathbf{W}\|_F^2 + \|\mathbf{P}_r\|_F^2\right), \end{aligned} \tag{37}$$

where $\mathbf{B} \in \{-1, 1\}^{k\times n}$. Matrix $\mathbf{X}_r$ represents the feature representations of the model input under multi-modalities.

In addition, optimizing Eq. (37) is analogous to the subsection of optimization algorithm. Accordingly, the solutions of $\mathbf{U}_r, \mathbf{W}, \mathbf{P}_r, \mathbf{R}$ can be computed via tuning the forms of Eqs. (17), (20), (23), (27), respectively. Meanwhile, we have the closed-form solution of $\mathbf{B}$ by adjusting Eqs. (33),(35),(36).

IV. Experiments

To validate the superiority of the proposed DJSAH, we perform a series of quantitative experiments on three commonly-used datasets over several state-of-the-art hashing methods.

A. Experimental Configuration

1) Datasets: We evaluate our method on three representative image-text search datasets, Wiki [59], Flickr25K [13] and NUS-WIDE [5]. Wiki possesses 2,866 image-text pairs with 10 class labels. Images are reduced as 128-dimensional SIFT feature vectors, and texts are extracted as 10-dimensional LDA topic vectors. Following in [9], 2,173 instances are randomly selected as the training samples and the rest as the querying samples. Flickr25K contains 25,000 images annotated with 24 semantic labels. The images are 512-dimensional GIST descriptors. The texts are converted to 1,386-dimensional BOW vector. Following in [15], we select 18,015 image-text data pairs as the training samples and the rest 2,000 instances as the querying samples. NUS-WIDE collects 269,648 images labeled with one of 81 semantic concepts. Following in [9], [41], the top 10 most frequent types (186,577 data pairs) are selected as the experimental data. The images are converted to 500-dimensional BOVW vector and The text are represented as 1,000-dimensional BOW vector. As [15] does, 2,000 data pairs are randomly obtained as the querying samples and 10,000 sampled data pairs used as the training samples.

2) Baselines and Evaluation Protocols: For fair comparison, several state-of-the-art approaches, including unsupervised hashing (LSSH [62], CMFH [9], JIMFH [44]), supervised hashing (SCM-seq [59], SePH [21], SMFH [23], LCMFH [41], SRLCH [36], SCRATCH [3], MSLF [45]), are chosen as the baseline methods for comparing with our DJSAH. We exploit the same training and query sets to evaluate the performance of these baselines. The experiments contain two kinds of cross-modal image-text search tasks: I→T and T→I, where the former represents that the images search for similar texts, and the latter denotes that the texts retrieve similar images. Following the protocols rigorously in [36], [44], three common evaluation metrics: mean Average Precision (**mAP**), precision-recall curve, and topN-precision curve are utilized in the later experiments. According to [36], [45], mAP can be computed as by

$$\mathrm{mAP} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{Z}\sum_{m=1}^{R} G_g(m)\xi(m), \tag{38}$$

where N is the number of querying points, Z represents the number of relevant points contained in searched results, $G(m)$ is the precision results of the top m searched points. In the following, we set the search points R as the size of the querying set for compare the mAP scores with the baselines. Besides, if mth point is relevant to $g(m)$, $\xi(m) = 1$; otherwise $\xi(m) = 0$. Generally, a larger score delivers better performance of cross-modal image-text hashing methods.

3) Implementation Details: In the experiments, the source codes of the baselines are provided by the kind authors. And we record their best results under three evaluation metrics by following the parameter settings in the original paper. For SRLCH, we independently complete it based on the paper. For DJSAH, we experimentally implement the parameter settings on three benchmark datasets. Specifically, the optimal result of DJSAH is obtained when parameters $\{\lambda_1 = 10^{-3}, \lambda_2 = 10^3, \lambda_3 = 10^{-3}, \lambda_4 = 10^{-3}, \theta = 10^{-1}\}$ on Wiki which has much fewer training instances. Moreover, we set $\{\lambda_1 = 10^{-1}, \lambda_2 = 10^5, \lambda_3 = 10^{-3}, \lambda_4 = 10^{-3}, \theta = 10^{-3}\}$ on larger Flickr25K and NUS-WIDE. Besides, the parameter sensitive analysis on all datasets are illustrated in the Subsection of parameter sensitivity. In the meanwhile, we set $w = 20$ by the later convergence analysis. The whole experiments are run 5 times for removing the impacts of randomness, then we average the results on all datasets. For fair comparison, all the methods are completed on the identical platform whose configuration is Intel(R) CPU @ 3.3 GHz, 10 cores, 128 GB memory.

B. Results and Analysis

1) Search Performance on mAP Metric: Table II lists the mAP results of all comparison method on all datasets with varied four code lengths. We can see that the proposed DJSAH works better than most baselines except for SCRATCH and MSLF in search performance with different code lengths and cross-modal datasets. Specifically, compared with the unsupervised hashing methods, the supervised baselines obtain better results. The reasonable explanation is that the latter

TABLE II
THE mAP SCORES OF ALL COMPARISON METHODS ON BENCHMARK DATASETS WITH VARYING CODE LENGTHS

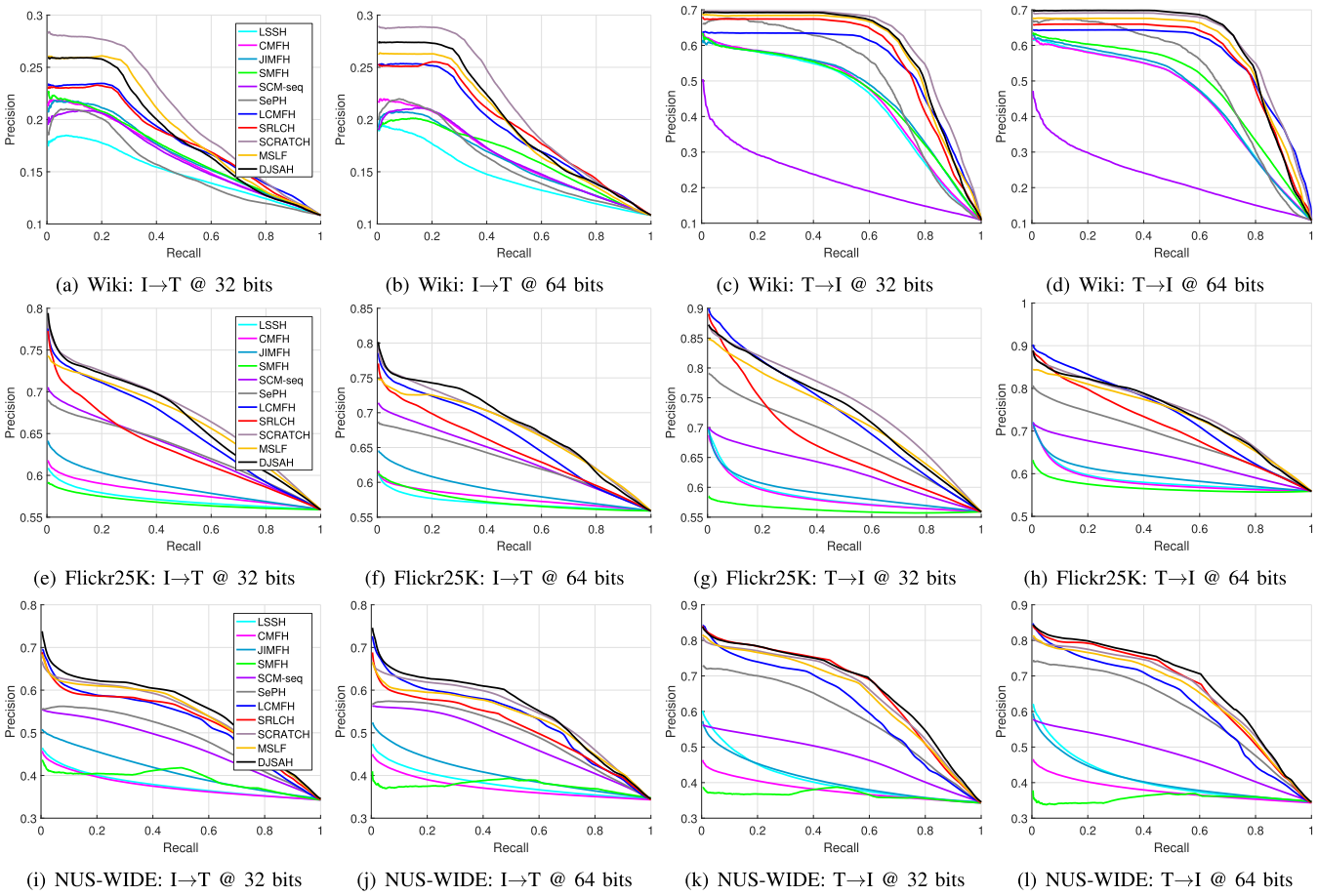
Task	Method	Wiki				Flickr25K				NUS-WIDE			
		16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
I→T	LSSH [62]	0.2252	0.2287	0.2373	0.2342	0.5733	0.5742	0.5736	0.5758	0.3900	0.3948	0.3993	0.3951
	CMFH [9]	0.2408	0.2570	0.2609	0.2578	0.5824	0.5821	0.5802	0.5779	0.3835	0.3807	0.3816	0.3827
	JIMFH [44]	0.2348	0.2429	0.2513	0.2520	0.5937	0.5908	0.5943	0.6007	0.4188	0.4200	0.4152	0.4201
	SMFH [23]	0.2284	0.2480	0.2670	0.2575	0.6116	0.6267	0.6414	0.6631	0.4633	0.4654	0.4504	0.4692
	SCM-Seq [59]	0.2471	0.2356	0.2393	0.2591	0.6241	0.6340	0.6432	0.6476	0.5739	0.5892	0.5718	0.6010
	SePH [21]	0.2484	0.2656	0.2565	0.2607	0.6542	0.6551	0.6564	0.6592	0.5481	0.5500	0.5627	0.5628
	LCMFH [41]	0.3222	0.3402	0.3519	0.3615	0.6807	0.6917	0.7016	0.7034	0.6096	0.6130	0.6213	0.6374
	SRLCH [36]	0.3307	0.3415	0.3633	0.3711	0.6571	0.6902	0.6891	0.6916	0.5933	0.6226	0.6349	0.6489
	SCRATCH [3]	**0.3605**	**0.3796**	**0.3886**	0.3914	**0.7126**	0.7144	0.7222	0.7282	0.6010	0.6303	0.6405	0.6460
	MSLF [45]	0.3324	0.3639	0.3751	**0.3995**	0.6959	0.7000	0.7159	0.7266	0.5950	0.6229	0.6254	0.6462
	DJSAH	0.3595	0.3615	0.3808	0.3935	0.6953	**0.7153**	**0.7327**	**0.7434**	**0.6190**	**0.6478**	**0.6492**	**0.6575**
T→I	LSSH [62]	0.6216	0.6352	0.6424	0.6354	0.5859	0.5882	0.5888	0.5876	0.4199	0.4216	0.4234	0.4198
	CMFH [9]	0.6161	0.6238	0.6394	0.6418	0.5877	0.5861	0.5855	0.5828	0.3905	0.3874	0.3922	0.3918
	JIMFH [44]	0.5188	0.5400	0.5523	0.5561	0.6060	0.6021	0.6023	0.6101	0.4310	0.4252	0.4211	0.4218
	SMFH [23]	0.5787	0.6390	0.6688	0.6682	0.6192	0.6384	0.6697	0.7018	0.4301	0.4275	0.4284	0.4233
	SCM-seq [59]	0.3821	0.4482	0.4417	0.4425	0.6379	0.6511	0.6585	0.6660	0.5621	0.5912	0.6035	0.6240
	SePH [21]	0.6786	0.6838	0.6982	0.6834	0.6952	0.7051	0.6998	0.7044	0.6369	0.6482	0.6674	0.6704
	LCMFH [41]	0.6972	0.7128	0.7269	0.7310	0.7361	0.7533	0.7745	0.7759	0.6921	0.7104	0.7197	0.7352
	SRLCH [36]	0.7154	0.7222	0.7454	0.7508	0.6969	0.7388	0.7466	0.7557	0.7233	0.7583	0.7631	0.7762
	SCRATCH [3]	0.7429	**0.7519**	0.7587	0.7579	**0.7692**	**0.7812**	**0.7902**	0.7969	0.7260	0.7517	0.7616	0.7618
	MSLF [45]	0.7211	0.7468	0.7467	0.7595	0.7454	0.7554	0.7796	0.7956	0.7076	0.7426	0.7515	0.7705
	DJSAH	**0.7430**	0.7509	**0.7603**	**0.7625**	0.7419	0.7651	0.7885	**0.8117**	**0.7390**	**0.7701**	**0.7807**	**0.7816**

can leverage the shared supervision information against the former in the training instances to enhance the quality of the unified hash codes. Moreover, our approach yields a significant improvement over the unsupervised hashing methods in the mAP metric. Clearly, we see from Table II that the average mAP of DJSAH is improved from 23.14% (LSSH) and 24.68% (JIMFH) to 37.38% on the Wiki datasets. Similar observations are also shown on the Flickr25K and NUS-WIDE datasets. Compared with several supervised approaches, DJSAH both performs better on two search tasks. For example, the mAP values of our method is 2.99% (Wiki), 2.73% (Flickr25K), 2.31% (NUS-WIDE) higher than LCMFH on the I→T task, and 3.36% (Wiki), 1.69% (Flickr25K), 5.35% (NUS-WIDE) higher than LCMFH on the T→I task. The main explanation may be that DJSAH fully utilizes multiple supervision including the class labels and pairwise similarities to yield the high-quality hash codes for training.

Moreover, compared to the best competitor SRLCH, the mAP scores of DJSAH is up to 2.22% (I→T) & 2.07%(T→I) for Wiki, 3.57% (I→T) & 4.23%(T→I) for Flickr25K and 1.84% (I→T) & 1.26%(T→I) for NUS-WIDE. Obviously, the proposed method has great improvement on the Wiki and Flickr25K datasets, but the improvement on the NUS-WIDE datasets is not obvious enough. Although obtaining comparable performance against the DJSAH on NUS-WIDE, SRLCH need more heavy training time than our method that reflected in Figure 5. This situation of SRLCH makes it impractical with applications to large-scale cross-modal image-text datasets. Additionally, as the code lengths increases, the mAP values of the baselines gradually increase, in which our method is always superior to others. Overall, these observations illustrate the effectiveness of the DJSAH with regard to mAP metric.

2) Search Performance on Precision-Recall Metric: Figure 3 shows the precision-recall curves among all comparison methods. Only 32 and 64 bits code lengths are reported for the limited space, in which DJSAH consistently obtains the best precision over the state-of-the-art baseline methods on three benchmark datasets. Actually, with the increasing of the recall, the precisions of all hashing methods are both in a downward trend. Further, DJSAH obtains much higher precision under different recall levels in terms of two code lengths, which further show the superior precision-recall scores of the proposed approach. In addition, the trends of precision-recall results are analogous to that of the mAP. In short, these above results explain the superiority of DJSAH under the precision-recall curve evaluation metric.

3) Search Performance on topN-Precision Metric: The topN-precision curves of all comparison methods are displayed in Figure 4 with fixed 32 and 64 bits. Clearly, we see that our DJSAH possesses the best precision against others on the Wiki and NUS-WIDE datasets. It indicates that our method yields better search performance by leveraging inner data characteristics and multiple semantic information. In addition, the precision scores of DJSAH are inferior to that of LCMFH, and comparable to that of SRLCH on Flickr25K while the number of retrieved points (a.k.a. N) are fixed. The possible



(a) Wiki: I→T @ 32 bits (b) Wiki: I→T @ 64 bits (c) Wiki: T→I @ 32 bits (d) Wiki: T→I @ 64 bits

(e) Flickr25K: I→T @ 32 bits (f) Flickr25K: I→T @ 64 bits (g) Flickr25K: T→I @ 32 bits (h) Flickr25K: T→I @ 64 bits

(i) NUS-WIDE: I→T @ 32 bits (j) NUS-WIDE: I→T @ 64 bits (k) NUS-WIDE: T→I @ 32 bits (l) NUS-WIDE: T→I @ 64 bits

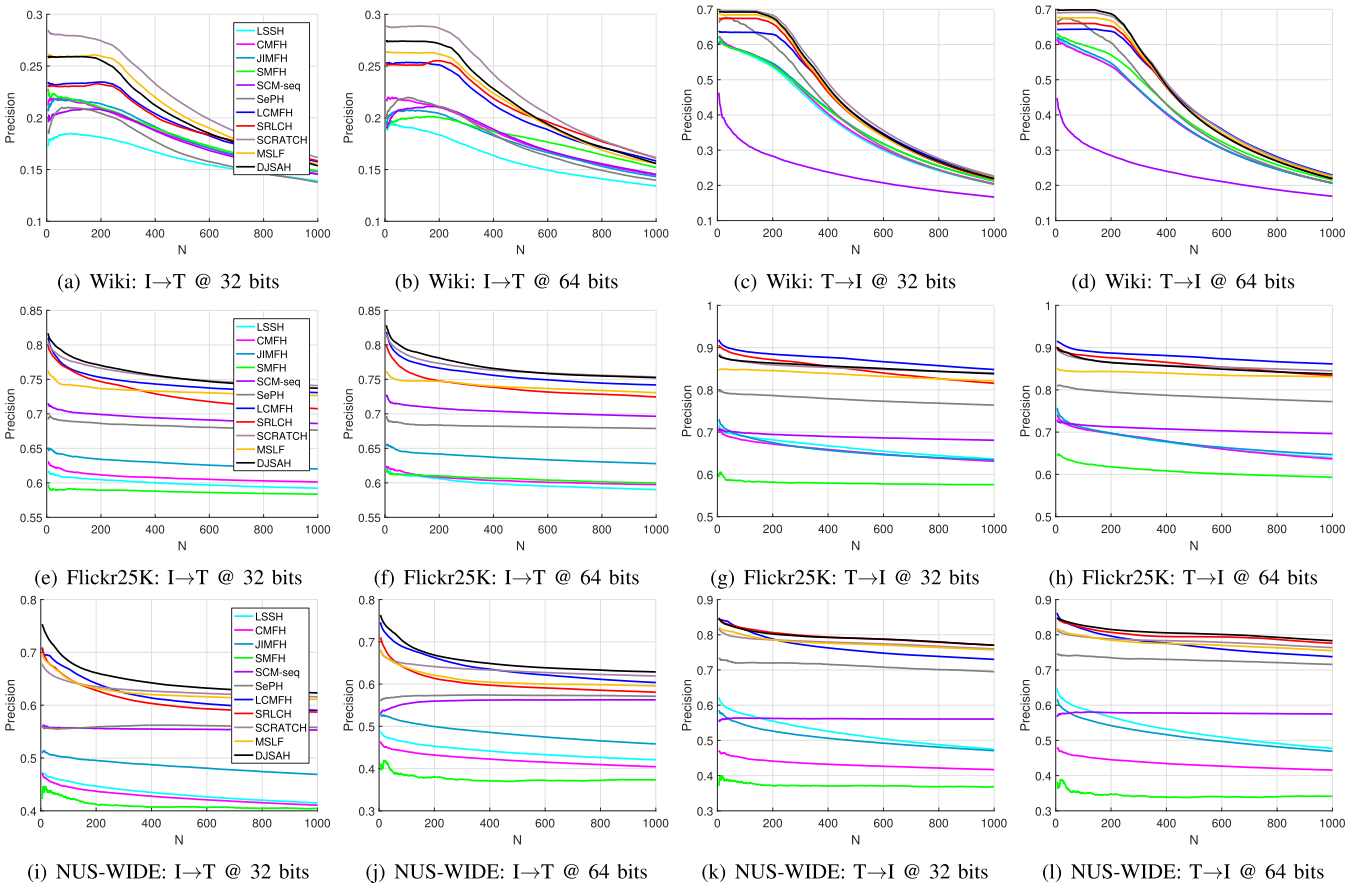
Fig. 3. The precision-recall curves of all comparison methods @ 32 bits and 64 bits on all datasets.

explanation may be that the ranking learning of LCMFH which strengthens the querying stage performs better than that of DJSAH. Besides, we observe that our method still delivers the best search precision (except for the tasks of T→I @ 32 and 64 bits) when the number of retrieved points are varied from 1 to 1,000. Therefore, it is clear that the DJSAH works better, which further proves the benefits of fully multiple shared semantics of data instances.

4) Training Cost: As stated above, the time and space complexity analysis of DJSAH has been proven in the Subsection of complexity analysis. In this part, we experimentally compare the training costs of DJSAH and the best two competitors on larger Flickr25K and NUS-WIDE, as is shown in Figure 5. It is clear that DJSAH outperforms SRLCH with lower training cost (except for LCMFH). Although DJSAH achieves the comparable training cost to LCMFH on two datasets, it has a substantial improvement of 3.15% against LCMFH on two search tasks. What's more, despite achieving similar search performance over our method on NUS-WIDE, the best competitor SRLCH consumes nearly 4-times of training cost at various code lengths. Generally, DJSAH can yield the best performance as well as efficiently train the proposed model. In addition, we compare the DJSAH over deep feature (named DJSAHcnn) with several deep hashing methods in supplementary material III. When performing training, DJSAHcnn takes 38.89, 46.46, 19.79 (in seconds) at 16, 32, and 64 hash bits, and the deep methods require more than two hours of training time. This is because our model inputs extracted handcrafted features, while deep hashing methods input raw images during training. These observations show that the proposed shallow DJSAH greatly improves its accuracy when using deep features as input. Moreover, our method achieves high-efficiency search applications while achieving comparable performance with existing deep hashing methods.

C. Model Analysis

1) Ablation Study: To further demonstrate the efficacy of the joint semantic alignment learning part and discrete optimization algorithm, we design the three variants of DJASH for comparison. As mentioned previously, the joint semantic alignment includes the class label alignment and the pairwise similarity alignment. The DJSAH-1 is the variant that employs the class labels to train objective ($\lambda_1 = 0$). The DJSAH-2 is the variant that integrates the pairwise similarities into the original instances (i.e., $\lambda_2 = 0$). The DJSAH-r is the variant by using the continuous relaxation strategy to perform the DJSAH, which relaxing the discrete condition as the continuous constraint during optimization. The performance comparison between DJSAH and its variants are reported in Table III. As expected, DJSAH performs the best than the other three variants on three datasets, that is, the optimal



(a) Wiki: I→T @ 32 bits (b) Wiki: I→T @ 64 bits (c) Wiki: T→I @ 32 bits (d) Wiki: T→I @ 64 bits

(e) Flickr25K: I→T @ 32 bits (f) Flickr25K: I→T @ 64 bits (g) Flickr25K: T→I @ 32 bits (h) Flickr25K: T→I @ 64 bits

(i) NUS-WIDE: I→T @ 32 bits (j) NUS-WIDE: I→T @ 64 bits (k) NUS-WIDE: T→I @ 32 bits (l) NUS-WIDE: T→I @ 64 bits

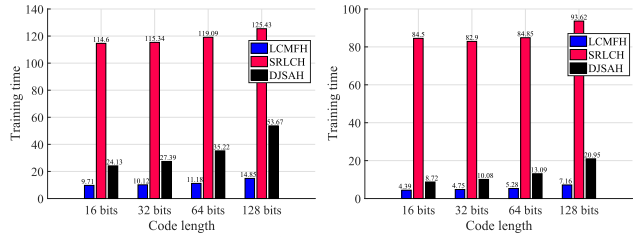
Fig. 4. The topN-precision curves of all comparison methods @ 32 bits and 64 bits on all datasets.

TABLE III
THE mAP RESULTS OF DJSAH AND ITS VARIANTS ON ALL DATASETS

Task	Method	Wiki				Flickr25K				NUS-WIDE			
		16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
I→T	DJSAH-1	0.3451	0.3587	0.3612	0.3872	0.6431	0.6503	0.6728	0.6736	0.5811	0.6043	0.6114	0.6262
	DJSAH-2	0.3357	0.3643	0.3650	0.3759	**0.7112**	0.7129	0.7172	0.7320	0.6068	0.6111	0.6422	0.6530
	DJSAH-r	0.3192	0.3526	0.3571	0.3722	0.6645	0.6778	0.6858	0.6965	0.5925	0.6053	0.6147	0.6285
	DJSAH	**0.3602**	**0.3622**	**0.3814**	**0.3938**	0.6947	**0.7152**	**0.7331**	**0.7443**	**0.6188**	**0.6481**	**0.6492**	**0.6583**
T→I	DJSAH-1	0.7286	0.7472	0.7570	0.7600	0.6677	0.6821	0.7164	0.7183	0.7052	0.7371	0.7582	0.7653
	DJSAH-2	0.7291	0.7313	0.7352	0.7447	0.7322	0.7586	0.7740	0.7908	0.7061	0.7112	0.7411	0.7702
	DJSAH-r	0.7184	0.7214	0.7409	0.7489	0.7321	0.7434	0.7688	0.7727	0.7044	0.7166	0.7385	0.7538
	DJSAH	**0.7432**	**0.7509**	**0.7613**	**0.7631**	**0.7421**	**0.7648**	**0.7890**	**0.8122**	**0.7387**	**0.7701**	**0.7812**	**0.7818**

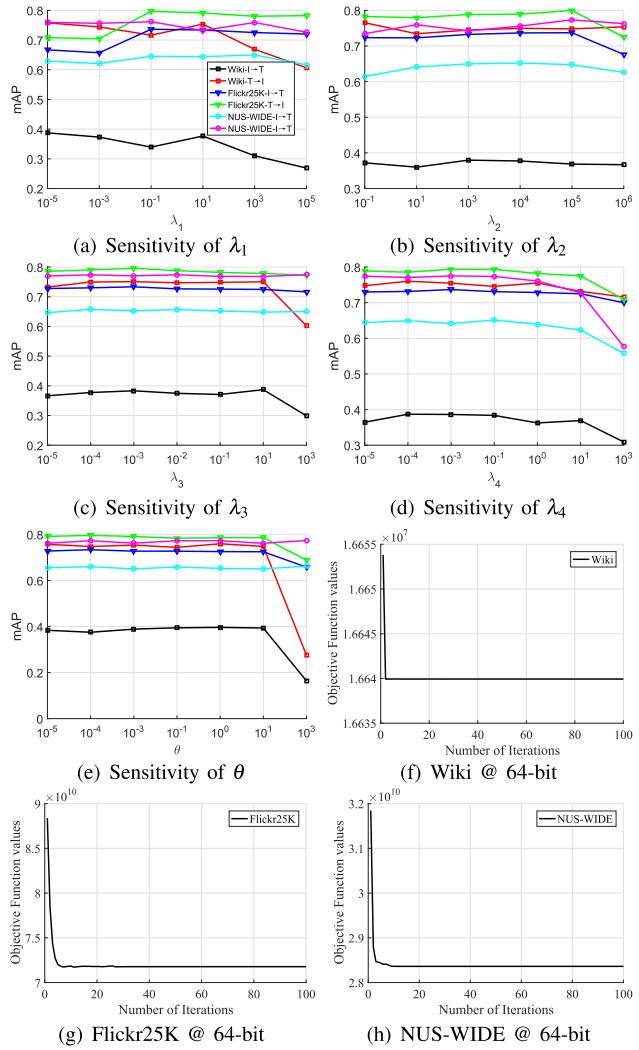
performance is achieved when the two constraint terms are available. Therefore, these results demonstrate that the complementary of the joint semantic alignment learning can enhance the performance. Besides, the above observation shows that the proposed discrete optimization algorithm works better than the traditional iteration strategy in improving the DJSAH model.

2) Parameter Sensitivity: Figures 6(a)-(e) report the performance of DJSAH with varying parameters at 64 hash bits (limited space) on all datasets. Among the parameters in Eq. (14), λ_1 weighs the contributions of learning hash codes by the pairwise similarities, λ_2 influences the weights of learning hash codes by the class labels, λ_3 measures the impact of learning hash functions, λ_4 denotes the regularization for averting the over-fitting problem and θ is the parameter for optimization algorithm. From Figures 6(a)-(e), we observe that DJSAH generally achieves good performance when $\lambda_1 \in [10^{-1}, 10]$, $\lambda_2 \in [10^4, 10^5]$, $\lambda_3 \in [10^{-3}, 10^{-1}]$, $\lambda_4 \in [10^{-3}, 10^{-1}]$ and $\theta \in [10^{-5}, 10^{-3}]$ on Flickr25K. Similar phenomenons can be found on Wiki and NUS-WIDE. Moreover, it is clear that the best performance of DJSAH is inconsistent under different datasets. And thus the parameter settings of DJSAH are varied on three cross-modal image-text datasets. It may be that



(a) Training time @ Flickr25K (b) Training time @ NUS-WIDE

Fig. 5. Training time comparisons of DJSAH and the best two competitors @ 64-bit on Flickr25K and NUS-WIDE.



(a) Sensitivity of λ_1 (b) Sensitivity of λ_2

(c) Sensitivity of λ_3 (d) Sensitivity of λ_4

(e) Sensitivity of θ (f) Wiki @ 64-bit

(g) Flickr25K @ 64-bit (h) NUS-WIDE @ 64-bit

Fig. 6. Model analysis of DJSAH @ 64-bit. (a)-(e) are parameter variations on Flickr25K. (f)-(h) is the convergence curve of objective function on all datasets.

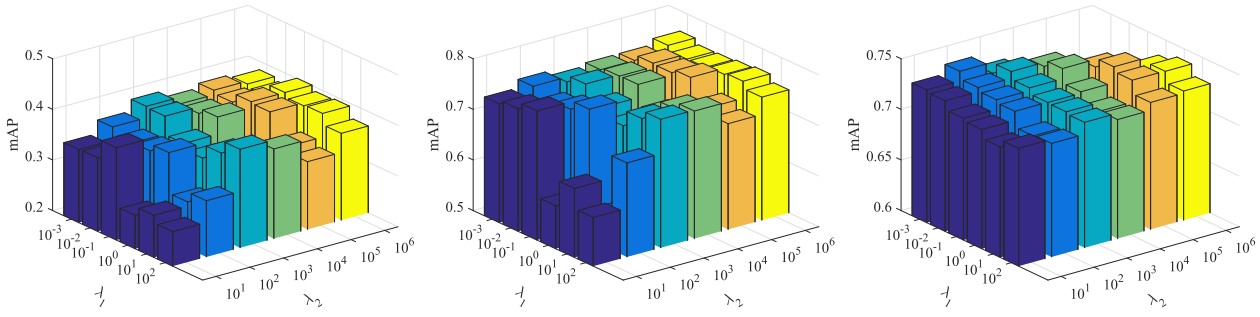
there are distinct extracted features and training data samples. In summary, DJSAH keep good and stable performance in the range of these parameters on all datasets.

Specifically, from Figures. (6)(a)-(d), we see that the mAP scores of our method drop significantly when the values of the parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ increased. The main reason may be that the proportion of the corresponding term increases, thus affecting the influence of each term in the objective function when the values of these parameters become larger. For example, the increasing influence of multiple supervised information items may lead to the weakening of the distribution structure information of the original data, and the "loss" of useful information ultimately reduces the search accuracy of the hashing model. The other possible explanation is that the larger parameters result in the imprecise hash codes. According to Eq. (33), we find that the result of hash code matrix $\mathbf{B}$ is mainly jointly determined by the parameters λ_1 and λ_2. The large value of these two parameters directly causes the closed-form solution of the hash code to be suboptimal and subsequent unsatisfactory search accuracy. Besides, we also assess the interaction of parameters λ_1 and λ_2 on search performance. Figure 7 shows the mAP results with varying parameters λ_1 and λ_2 on all datasets. From Figure 7, we see that DJSAH delivers stable performance over $\lambda_1 \in [10^{-3}, 10^{-1}]$ and $\lambda_2 \in [10^3, 10^5]$ on the Wiki datasets, $\lambda_1 \in [10^{-1}, 10^0]$ and $\lambda_2 \in [10^4, 10^5]$ on the larger Flickr25K and NUS-WIDE datasets, respectively. The search performance of the DJSAH become sensitive and poor when the parameters of these two parameters are not in the corresponding value ranges. Thus, to possess better results, we tune the influence of λ_1 and λ_2 by ourselves. It indicates that our method can produce reasonable and robust results when keeping the balance of λ_1 and λ_2 and meanwhile having a insensitivity region to the other parameters.

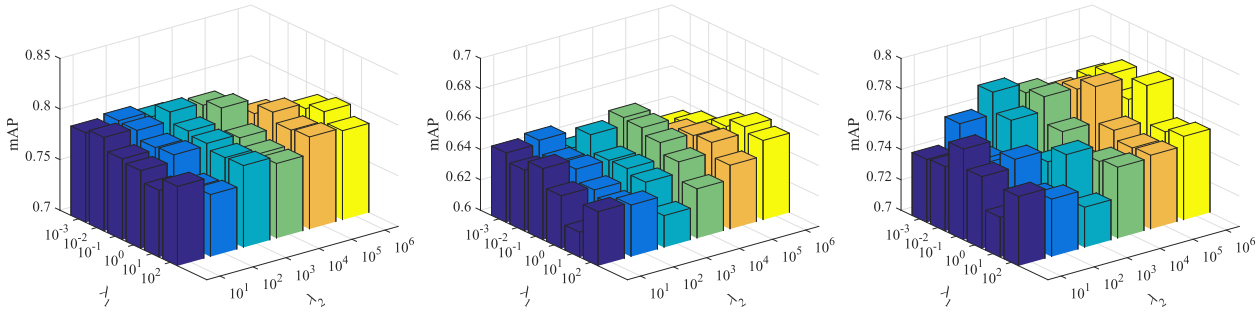
3) Convergence Analysis: To validate the feasibility of the proposed DJSAH, we conduct the convergence experiment at 64-bit hash code length on all datasets, which is plotted in Figures 6(f)-(h). We compute the objective function values in the formulation of Eq. (14) with the varying iteration numbers w in Algorithm 1. To be specific, as the number of iterations increases, the objective function value of DJSAH decreases very fast on Wiki and tends to keep stable less than 20 iterations on larger Flick25K and NUS-WIDE. We have proved the convergence of the proposed discrete optimization algorithm in supplementary material II. Meanwhile, we also perform the corresponding convergence analysis on 16 and 128 code lengths on Wiki and NUS-WIDE. As is reported in Figure 8, we see that the optimization algorithm posses a fast convergence within 20 iteration numbers. These observations conform the proposed algorithm can converge in quite fast training speed, which further illustrate the satisfactory convergence efficiency of our DJSAH.

D. Failure Examples and Follow-up Study

Despite working better than several existing hashing methods on three popular datasets, the proposed DJSAH may suffer from the unsatisfactory ranking situations in the above experiments. The failure examples are illustrated in Figures 4(g)-(h). For instance, the precisions of our DJSAH are weaker than the SRLCH while being comparable to the LCMFH at 32 and 64 hash bits under the topN-precision metric for the T→I task. A possible explanation could be that our approach does not well handle the raking information during training, that is, with the increasing of the number of searched samples, several levels of the precision make a small deviation because of the

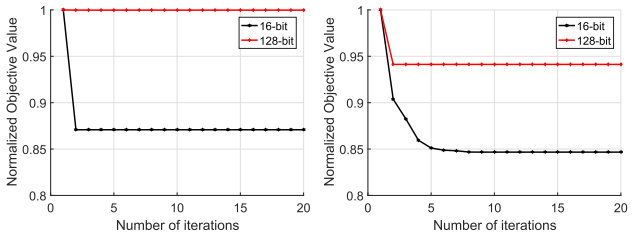


(a) Wiki: I→T @ 64bits (b) Wiki: T→I @ 64bits (c) Flickr25K: I→T @ 64bits



(d) Flickr25K: T→I @ 64bits (e) NUS-WIDE: I→T @ 64bits (f) NUS-WIDE: T→I @ 64bits

Fig. 7. mAP results with varying parameters λ_1 and λ_2 on three public datasets.



(a) Convergence @ Wiki (b) Convergence @ NUS-WIDE

Fig. 8. The convergence curves of the proposed discrete optimization algorithm.

introduction of redundant information of input feature representations. consequently, it brings the dissimilar image-text data pairs during learning to rank, which produces the bad cases in the querying stage.

In view of the above analysis, some topN-precision scores of our approach can be further promoted by the following directions: First, a useful learning to rank scheme is introduced to handle more complex semantic similar of training instances. Second, we plan to incorporate the ranking information as the supervised information to map into the shared common space to obtain higher quality hash codes for the search tasks. In addition, since multi-modal information retrieval is one of the long-term development hotspots in the future, we intend to obtain the common feature representation of the identical data instance for the direction of multi-modal hashing search.

V. Conclusion

This paper proposes a novel Discrete Joint Semantic Alignment Hashing (DJSAH) method with applications to cross-modal image-text search. Concretely, we formulate a joint semantic alignment model to integrate the multiple high-level semantics including class labels and the linear pairwise similarity matrix to gain the discriminative capability of the to-be-learned hash codes. In the meanwhile, the low-level data characteristics of original instances are also taken into account. To reduce the large quantization information errors in the traditional continuous relaxation strategy, an optimization algorithm with linear computation and memory cost is developed to directly obtain the discrete hash codes. Comprehensive experiments on three frequently-used datasets show that DJSAH achieves the state-of-the-art search results. In future, we will design a more comprehensive feature description framework by leveraging the complementarity between data modalities including the spatial information of the image and the semantic attributes of the text to improve its performance.

References

[1] H. Chen, G. Ding, Z. Lin, S. Zhao, and J. Han, "Cross-modal image-text retrieval with semantic consistency," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1749–1757.

[2] H. Chen, G. Ding, X. Liu, Z. Lin, J. Liu, and J. Han, "IMRAM: Iterative matching with recurrent attention memory for cross-modal image-text retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12652–12660.

[3] Z.-D. Chen, C.-X. Li, X. Luo, L. Nie, W. Zhang, and X.-S. Xu, "SCRATCH: A scalable discrete matrix factorization hashing framework for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2262–2275, Jul. 2020.

[4] Z.-D. Chen, Y. Wang, H.-Q. Li, X. Luo, L. Nie, and X.-S. Xu, "A two-step cross-modal hashing by exploiting label correlations and preserving similarity in both steps," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1694–1702.

[5] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of Singapore," in *Proc. ACM Int. Conf. Image Video Retr. (CIVR)*, 2009, pp. 48–56.

[6] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 1271–1284, 2020.

[7] A. Daniilidis, D. Drusvyatskiy, and A. S. Lewis, "Orthogonal invariance and identifiability," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 2, pp. 580–598, Jan. 2014.

[8] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3893–3903, Aug. 2018.

[9] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2083–2090.

[10] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 817–824.

[11] H.-J. Huang, R. Yang, C.-X. Li, Y. Shi, S. Guo, and X.-S. Xu, "Supervised cross-modal hashing without relaxation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1159–1164.

[12] Y. Huang and L. Wang, "ACMM: Aligned cross-modal memory for few-shot image and sentence matching," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5773–5782.

[13] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr. (MIR)*, 2008, pp. 39–43.

[14] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3270–3278.

[15] Q.-Y. Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3490–3501, Jul. 2019.

[16] L. Jin, K. Li, Z. Li, F. Xiao, G.-J. Qi, and J. Tang, "Deep semantic-preserving ordinal hashing for cross-modal similarity search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1429–1440, May 2019.

[17] Q. Kong, Z. Wu, Z. Deng, M. Klinkigt, B. Tong, and T. Murakami, "MMAct: A large-scale dataset for cross modal human action understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8657–8666.

[18] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1360–1365.

[19] K.-H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. Eur. Conf. Artif. Intell.*, 2018, pp. 212–228.

[20] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4242–4251.

[21] Z. Lin, G. Ding, J. Han, and J. Wang, "Cross-view retrieval via probability-based semantics-preserving hashing," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4342–4355, Dec. 2017.

[22] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.

[23] H. Liu, R. Ji, Y. Wu, and G. Hua, "Supervised matrix factorization for cross-modality hashing," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1767–1773.

[24] J. Liu, M. Yang, C. Li, and R. Xu, "Improving cross-modal image-text retrieval with teacher-student learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3242–3253, Aug. 2021.

[25] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2074–2081.

[26] X. Lu, L. Zhu, Z. Cheng, L. Nie, and H. Zhang, "Online multi-modal hashing with dynamic query-adaption," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 715–724.

[27] J. Luo, Y. Shen, X. Ao, Z. Zhao, and M. Yang, "Cross-modal image-text retrieval with multitask learning," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2309–2312.

[28] J. Luo, Y. Wo, B. Wu, and G. Han, "Learning sufficient scene representation for unsupervised cross-modal retrieval," *Neurocomputing*, vol. 461, pp. 404–418, Oct. 2021.

[29] X. Luo, L. Nie, X. He, Y. Wu, Z.-D. Chen, and X.-S. Xu, "Fast scalable supervised hashing," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 735–744.

[30] X. Ma, T. Zhang, and C. Xu, "Multi-level correlation adversarial hashing for cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 12, pp. 3101–3114, Dec. 2020.

[31] X. Nie, X. Liu, X. Xi, C. Li, and Y. Yin, "Fast unmediated hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 9, pp. 3669–3678, Sep. 2021.

[32] X. Nie *et al.*, "Deep multiscale fusion hashing for cross-modal retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 401–410, Jan. 2021.

[33] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, "Hierarchical multimodal LSTM for dense visual-semantic embedding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1899–1907.

[34] Y. Peng, X. Huang, and Y. Zhao, "An overview of cross-media retrieval: Concepts, methodologies, benchmarks, and challenges," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2372–2385, Sep. 2018.

[35] J. Qin *et al.*, "Discrete semantic matrix factorization hashing for cross-modal retrieval," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 1550–1557.

[36] H. T. Shen *et al.*, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3351–3365, Oct. 2021.

[37] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. Int. Conf. Manage. Data (SIGMOD)*, 2013, pp. 785–796.

[38] S. Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3027–3035.

[39] J. Tang, K. Wang, and L. Shao, "Supervised matrix factorization hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3157–3166, Jul. 2016.

[40] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multimodal hashing for cross-media retrieval," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3890–3896.

[41] D. Wang, X.-B. Gao, X. Wang, and L. He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2466–2479, Oct. 2018.

[42] D. Wang, Q. Wang, Y. An, X. Gao, and Y. Tian, "Online collective matrix factorization hashing for large-scale cross-media retrieval," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1409–1418.

[43] D. Wang, Q. Wang, and X. Gao, "Robust and flexible discrete hashing for cross-modal similarity search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2703–2715, Oct. 2018.

[44] D. Wang, Q. Wang, L. He, X. Gao, and Y. Tian, "Joint and individual matrix factorization hashing for large-scale cross-modal retrieval," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107479.

[45] S. Wang, H. Zhao, and K. Nai, "Learning a maximized shared latent factor for cross-modal hashing," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107252.

[46] Y. Wang, X. Ou, J. Liang, and Z. Sun, "Deep semantic reconstruction hashing for similarity retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 387–400, Jan. 2021.

[47] Y. Wang and Y. Peng, "MARS: Learning modality-agnostic representation for scalable cross-media retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Dec. 16, 2021, doi: 10.1109/TCSVT.2021.3136330.

[48] Z. Wang *et al.*, "CAMP: Cross-modal adaptive message passing for text-image retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5763–5772.

[49] X. Wen, Z. Han, and Y.-S. Liu, "CMPD: Using cross memory network with pair discrimination for image-text retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2427–2437, Jun. 2021.

[50] F. Wu *et al.*, "Supervised discrete matrix factorization hashing for cross-modal retrieval," in *Proc. IEEE Int. Conf. Cloud Comput. Intell. Syst.*, Nov. 2018, pp. 855–859.

[51] Y. Wu, X. Luo, X.-S. Xu, S. Guo, and Y. Shi, "Dictionary learning based supervised discrete hashing for cross-media retrieval," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2018, pp. 222–230.

[52] Z. Wu, J. Li, J. Xu, and W. Yang, "Beyond ITQ: Efficient binary multi-view subspace learning for instance retrieval," *J. Vis. Commun. Image Represent.*, vol. 79, Aug. 2021, Art. no. 103234.

[53] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2156–2162.

[54] X. Xu, F. Shen, Y. Yang, and H. T. Shen, "Discriminant cross-modal hashing," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2016, pp. 305–308.

[55] X. Xu, T. Wang, Y. Yang, L. Zuo, F. Shen, and H. T. Shen, "Cross-modal attention with semantic consistence for image-text matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5412–5425, Dec. 2020.
[56] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1618–1625.
[57] T. Yao *et al.*, "Efficient discrete supervised hashing for large-scale cross-modal retrieval," *Neurocomputing*, vol. 385, pp. 358–367, Apr. 2020.
[58] T. Yao, X. Kong, H. Fu, and Q. Tian, "Discrete semantic alignment hashing for cross-media retrieval," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4896–4907, Dec. 2020.
[59] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.
[60] J. Zhang and Y. Peng, "SSDH: Semi-supervised deep hashing for large scale image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 212–225, Jan. 2016.
[61] D. C. Zhen, J. Y. Wan, X. L. Chuan, L. Nie, and S. X. Xin, "Dual deep neural networks cross-modal hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 274–281.
[62] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2014, pp. 415–424.
[63] J. Zhou, G. Ding, Y. Guo, Q. Liu, and X. Dong, "Kernel-based supervised hashing for cross-view similarity search," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2014, pp. 1–6.
[64] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Deep collaborative multi-view hashing for large-scale image search," *IEEE Trans. Image Process.*, vol. 29, pp. 4643–4655, 2020.
[65] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. 21st ACM Int. Conf. Multimedia*, Oct. 2013, pp. 143–152.

Song Wang received the M.S. degree from the Changsha University of Science and Technology, China, in 2017. He is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China. His research interests include image processing, multimedia analysis and retrieval, pattern recognition, and computer vision.

Huan Zhao received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, Changsha, China, in 1989, 2004, and 2010, respectively. Currently, she is a Professor with the School of Information Science and Technology, Hunan University. She has published over 100 research papers in international journals and conferences, including *Information Processing and Management*, *Knowledge-based Systems*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IEEE International Conference on Acoustics, Speech and Signal Processing. Her current research interests mainly include speech signal processing, cross-media retrieval, and natural language processing.

Keqin Li (Fellow, IEEE) is a SUNY Distinguished Professor of computer science with the State University of New York. He is also a National Distinguished Professor with Hunan University, China. He has authored or coauthored over 840 journal articles, book chapters, and refereed conference papers. He holds over 70 patents announced or authorized by the Chinese National Intellectual Property Administration. His current research interests include cloud computing, fog computing, mobile edge computing, energy-efficient computing and communication, embedded systems, cyber-physical systems, heterogeneous computing systems, big data computing, high-performance computing, CPU-GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, and intelligent and soft computing. He is among the world's top five most influential scientists in parallel and distributed computing in terms of both single-year impact and career-long impact based on a composite indicator of Scopus citation database. He is also a member of Academia Europaea (The Academy of Europe). He was a recipient of several best paper awards. He has chaired many international conferences. He is currently an Associate Editor of the ACM *Computing Surveys* and the CCF *Transactions on High Performance Computing*. He was on the Editorial Boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.