Enhanced Discrete Multi-modal Hashing: More Constraints yet Less Time to Learn

Yong Chen, Hui Zhang, Zhibao Tian, Jun Wang, Dell Zhang, *Senior Member, IEEE*, and Xuelong Li, *Fellow, IEEE*

Abstract—Due to the exponential growth of multimedia data, multi-modal hashing as a promising technique to make cross-view retrieval scalable is attracting more and more attention. However, most of the existing multi-modal hashing methods either divide the learning process unnaturally into two separate stages or treat the discrete optimization problem simplistically as a continuous one, which leads to suboptimal results. Recently, a few discrete multi-modal hashing methods that try to address such issues have emerged, but they still ignore several important discrete constraints (such as the balance and decorrelation of hash bits). In this paper, we overcome those limitations by proposing a novel method named "Enhanced Discrete Multi-modal Hashing (EDMH)" which learns binary codes and hashing functions simultaneously from the pairwise similarity matrix of data, under the aforementioned discrete constraints. Although the model of EDMH looks a lot more complex than the other models for multi-modal hashing, we are actually able to develop a fast iterative learning algorithm for it, since the subproblems of its optimization all have closed-form solutions after introducing two auxiliary variables. Our experimental results on three real-world datasets have revealed the usefulness of those previously ignored discrete constraints and demonstrated that EDMH not only performs much better than state-of-the-art competitors according to several retrieval metrics but also runs much faster than most of them.

Index Terms—Learning to Hash, Discrete Optimization, Semantics Alignment, Cross-View Retrieval.

1 Introduction

Recently, abundant multimedia data, e.g., images, texts, and videos, have flooded people's lives [1], [2], [3], [4], [5], [6], [7], [8], which generates a huge demand for scalable cross-view retrieval techniques. Specifically, given one view of a query (such as a text query), users expect to find semantic related results not only that original view but also in other different views (such as images and videos). Multi-modal hashing (MH) for approximate nearest neighbor search holds the potential to handle such cross-view retrieval tasks on web-scale data, due to the binary codes which merely require economical storage resources and greatly accelerate the retrieval process with hardware-level XOR operations [9], [10], [11], [12].

Roughly speaking, MH methods could be divided into two categories: unsupervised and supervised.

Unsupervised MH methods usually focus on the intra- and inter-relationships of data points just with features in different modalities for hash codes and functions. Inter-Media Hashing (IMH) [13] explores the intra-view and inter-view correlations among multiple media types and transforms cross-modal instances into one common Hamming space. However, IMH needs to construct similarity matrix with a large computational complexity O(n^2) (n is the number of instances in dataset), which obviously obstacles its applications on large-scale databases. Fusion Similarity Hashing (FSH) [14], slightly different from IMH, learns unified binary codes in consistence with the self-defined fusion similarities across modalities, which still confronts the time-consuming cross-modal similarities, especially for large datasets. Collective Matrix Factorization Hashing (CMFH) [15] learns unified hash codes for instances via matrix factorizations with latent factor models from different modalities. Latent Semantic Sparse Hashing (LSSH) [16] obtains latent semantic components for images and texts with sparse coding and matrix factorization respectively, and then maps the learned features into a joint abstraction space. Semantic Topic Multi-modal Hashing (STMH) [17] first captures topics of texts and concepts of images via clustering techniques and robust matrix factorizations respectively, and then transforms the learned multi-modal features into a common subspace by their correlations. These three approaches (i.e., CMFH, LSSH and STMH) could be quite efficient in hashing learning and achieve satisfactory cross-view retrieval performances on scalable datasets. However, they don't fully utilize the supervised information, and relax the binary hashing problems as continuous optimizations, which would lead to sub-optimal solutions and therefore yield not the best binary codes for cross-view retrieval tasks. In addition, it's worth mentioning that the Collective Reconstructive Embeddings for Cross-Modal Hashing approach (CRE) [18], lately proposed, starts to explore complex constraints, such as balance and decorrelation of the to-be-learnt binary codes. Nevertheless, this method bridges the cross-modal semantic gaps via image-text pairs instead of the pairwise similarities across multi-modalities, which still exists a large space to further leveraging the cross-modal semantics for better hashing. Moreover, the binary constrained optimization problem, addressed

- *Yong Chen is with the Key Lab of Machine Perception, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (E-mail: butterfly.chinese@pku.edu.cn).*
- *Hui Zhang and Zhibao Tian are with the Department of Computer Science and Engineering, Beihang University, Beijing 100191, China.*
- *Jun Wang is a Professor at University College London, UK.*
- *Dell Zhang is the corresponding author (E-mail: dell.z@ieee.org). He is currently on leave from Birkbeck, University of London and working for Blue Prism AI Labs.*
- *Xuelong Li is with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China (E-mail: xuelong_li@nwpu.edu.cn).*

with the iterative Minorization-Maximum algorithm [19], could be transformed to a much simpler optimization problem with a faster closed-form solution.

Supervised Multi-modal Hashing (SMH) often preserves the pairwise similarities between different modalities in accordance with semantic labels for cross-view hash codes. Cross-modality Metric Learning using Similarity-Sensitive Hashing (CMSSH) [20] models the mapping from the original instances with cross-modalities into the shared Hamming space as binary classifiers, and learns them efficiently via boosting algorithms. Cross-View Similarity Search (CVH) [21] learns hash functions by minimizing the weighted Hamming distances between the hash codes of training samples. Semantic Correlation Maximization (SCM) [22] seamlessly integrates marked labels into the hashing learning procedure via maximizing semantic correlations for large-scale multi-modal retrieval. Semantics-Preserving Hashing (SePH) [23] first transforms the semantic affinities into a probability distribution and approximates it with be-learnt hash codes in Hamming space, and then learns the hash functions as a kernel logistic regression for each view. Generalized Semantic Preserving Hashing (GSPH) [24], [25] first learns the optimum hash codes for two modalities simultaneously, and then learns the hash functions to map the features to the hash codes. These supervised methods make full use of the supervised information and often outperform the above unsupervised approaches in cross-view retrieval missions. However, they also share the similar problems with the unsupervised methods, such as time-consuming similarity matrix construction and two-stage learning procedure (e.g., SePH and GSPH), relaxation from discrete to continuous (e.g., CMSSH, CVH, SCM, etc.), which indeed simplify the complex discrete optimization problems for binary codes and hash functions but meanwhile deteriorate the cross-view retrieval performances.

Very recently, there emerges several discrete SMH methods. Learning Discriminative Binary Codes for Cross-modal Hashing (DCH) [26] pursues discriminative binary codes by leveraging pointwise supervised class labels while keeping the discrete constraints. Discrete Manifold-Embedded Hashing (SDMCH) [27] first learns the local manifold structures via LLE [28], and then combines it with class label supervised binary hashing. Discrete Matrix Factorization Hashing (SCRATCH) [29] mainly compromises the merits of CMFH [15] and DCH [26], and develops a fast discrete hashing for cross-modal retrieval. Asymmetric Discrete Cross-Modal Hashing (ADCH) [30] integrates collective matrix factorization (CMF) with pairwise similarity supervised hashing in an asymmetric way. Discrete Latent Factor Hashing (DLFH) [31] tries to maximize the likelihood of the cross-modal data with pairwise simialrity maintained, and then solves the discrete constrained optimization by column-wise learning strategy. Other examples of the discrete SMH family include Robust Discrete Code Modeling (RDCM) [32] and Subspace Relation Learning for Cross-modal Hashing (SRLCH) [33]. Those methods all just utilize the simplest binary constraints for fast learning (usually neglecting some important constraints, such as the balance and decorrelation of hash codes), and thus it still exists great potentials for improvements.

By the way, there also have sprung up some deep MH models: Dual Deep Neural Networks Cross-Modal Hashing [34], Pairwise Relationship Guided Deep Hashing for Cross-Modal Retrieval [35], Deep Cross-Modal Hashing [36], Deep Binary Reconstruction for Cross-Modal Hashing [37], Triplet-Based Deep Hashing Network for Cross-Modal Retrieval [38], Deep Supervised Cross-Modal Retrieval [39], Unsupervised Deep Hashing with Similarity-Adaptive and Discrete Optimization [40], etc. Those deep-learning based methods can usually achieve a high accuracy for cross-modal retrieval, but they all require a very large amount of data (e.g., ImageNet) and take long time to train even with GPUs/TPUs. Quite the contrary, the focus of this paper is on low-cost super-fast MH without relying on expensive computational resources, which is still an open challenge in scalable cross-view retrieval.

Table 1
The notations used in this paper.

Symbol	Explanation
n	a scalar
$\boldsymbol{v}$	a vector
$\mathbf{M}$	a matrix
v_i	a scalar: the (i)-th element of vector $\boldsymbol{v}$
$\boldsymbol{m_i}$	a vector: the (i)-th column of matrix $\boldsymbol{M}$
m_{ij}	a scalar: the (ij)-th element of matrix $\boldsymbol{M}$
$\mathbf{0}_n$	an $n \times 1$ vector with all 0 elements
$\mathbf{1}_n$	an $n \times 1$ vector with all 1 elements
$\mathbf{I}_n$	an $n \times n$ identity matrix
$\mathbf{O}$	a matrix with all 0 elements
$\mathbf{M}^T$	the transpose of matrix $\boldsymbol{M}$
$\mathbf{D}^{-1}$	the inverse of square matrix $\mathbf{D}$
$tr(\mathbf{D})$	the trace of square matrix $\mathbf{D}$: $\sum_i d_{ii}$
$\|\|\boldsymbol{v}\|\|_2$	the l_2 norm of vector $\boldsymbol{v}$: $\sqrt{\sum_i v_i^2}$
$\|\|\mathbf{M}\|\|_F$	the Frobenius norm of $\mathbf{M}$: $\sqrt{\sum_{ij} m_{ij}^2}$
$\mathrm{sgn}(\cdot)$	the element-wise sign function

To further unleash the full potentials of low-cost non-deep-learning techniques for fast MH, we propose a novel SMH method, called "Enhanced Discrete Multi-modal Hashing (EDMH)", which seamlessly integrates semantic supervised hashing with complex beneficial constraints for retrieval. The main contributions can be listed as follows:

- Unlike the previous MH methods, a joint hashing learning model with three discrete constraints (i.e., binary values, balance codings and decorrelation of hash bits), which have not been addressed in discrete MH before, is proposed here for scalable cross-view retrieval tasks.
- Although such constraints make EDMH more complex and challenging to handle, two intermediate variables are introduced to convert EDMH into a simpler optimization problem, which contributes to the closed-form solutions for its subproblems and surprisingly makes the whole learning much faster than many baseline models.
- Experiments on three benchmark datasets exhibit that EDMH can not only outperform many state-of-the-art competitors in cross-view retrieval tasks, but also be fast-efficient on scalable scenarios, e.g., NUS-WIDE. In addition, a comparative experiment is carefully designed to demonstrate the obvious performance gain by adding "balance and decorrelation" constraints into the only-binary-values constrained MH.

2 Problem Statement

Multi-modal hashing aims to build up the connections between different modalities, and then benefits cross-view retrievals. For easier presentation, we describe the SMH problem with only two modalities (e.g., images and texts), because it could be easily extended to multiple modalities. For more details about the general extensions, please refer to Section 5.

Given m images and n texts associated with shared tags, they can be represented as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m]^T \in \mathbb{R}^{m \times d_x}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]^T \in \mathbb{R}^{n \times d_y}$ respectively, where d_x and d_y correspond to the dimensions of their feature spaces. Set $\mathbf{G}_x \in \{0,1\}^{m \times l}$ and $\mathbf{G}_y \in \{0,1\}^{n \times l}$ to be the label matrices of images $\mathbf{X}$ and texts $\mathbf{Y}$ respectively, where 1(0) denotes that the image/text has(not) the tag, and l is the number of shared labels. Note that the pairwise similarity $\mathbf{S}_{xy}$ between images and texts could be calculated in some function with $\mathbf{G}_x\mathbf{G}_y^T$ as parameter. The goal of SMH is to learn two hashing functions: $\mathbf{H}_x(\mathbf{x}) : \mathbb{R}^{d_x} \to \{-1,+1\}^q$ and $\mathbf{H}_y(\mathbf{y}) : \mathbb{R}^{d_y} \to \{-1,+1\}^q$, where q is the length of binary hash codes. Most importantly, the hash functions could map the original feature vectors in each modality into a common Hamming space which should preserve the pairwise semantics in accordance with marked tags, i.e., if $\mathbf{x}_i$ and $\mathbf{y}_j$ share more common labels, then the Hamming distance between their binary codes should be smaller; and vice versa. Here, we adopt the simplest linear projections for hash functions, which are defined as follows:

$$\mathbf{B}_x = \mathbf{H}_x(\mathbf{X}) = \mathrm{sgn}(\mathbf{X}\mathbf{W}_x) \ , \tag{1}$$

$$\mathbf{B}_y = \mathbf{H}_y(\mathbf{Y}) = \mathrm{sgn}(\mathbf{Y}\mathbf{W}_y) \ , \tag{2}$$

where $\mathrm{sgn}(\cdot)$ is an element-wise sign function that outputs $+1$ if the input value is non-negative and -1 otherwise. $\mathbf{W}_x \in \mathbb{R}^{d_x \times q}$ and $\mathbf{W}_y \in \mathbb{R}^{d_y \times q}$ correspond to the projections for hash functions $\mathbf{H}_x$ and $\mathbf{H}_y$, $\mathbf{B}_x \in \{-1,+1\}^{m \times q}$ and $\mathbf{B}_y \in \{-1,+1\}^{n \times q}$ are the binary codes for images and texts respectively. Therefore, the SMH problem is formally to learn $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{W}_x$ and $\mathbf{W}_y$ from the data matrices $\mathbf{X}$ and $\mathbf{Y}$, and the label matrices $\mathbf{G}_x$ and $\mathbf{G}_y$. Note that although SMH is described with general forms, we would mainly focus on the paired image-text instances (i.e., $m = n$ and $\mathbf{G}_x = \mathbf{G}_y$) because they could be easily obtained in the real-world scenarios, such as Wikipedia (images surrounded by texts) and Flickr (images with marked tags).

As a convention, we use boldface uppercase letters like $\mathbf{M}$ to denote matrices, and boldface lowercase letters like $\mathbf{v}$ to mark vectors. The (ij)-th element of matrix $\mathbf{M}$ is m_{ij} and the Frobenius norm is then defined as $||\mathbf{M}||_F = \sqrt{\sum_{ij} m_{ij}^2}$. The boldface $\mathbf{1}_n$ and $\mathbf{0}_n$ correspond to vectors sized by $n \times 1$ with all 1-elements and 0-elements, respectively. Similarly, the boldface $\mathbf{O}$ is a matrix with all 0-elements. $\mathbf{I}_d$ is an identity matrix sized by $d \times d$. Besides, for a square matrix $\mathbf{D} = (d_{ij})_{m \times m} \in \mathbb{R}^{m \times m}$, the trace function is defined as $tr(\mathbf{D}) = \sum_{i=1}^{m} d_{ii}$. To the end, Table 1 gives a brief summary of the adopted notations in this paper.

3 Proposed Method

Here we describe in detail EDMH, an enhanced supervised discrete multi-modal hashing method in the joint learning framework which simultaneously obtains binary codes and hash functions.

3.1 Similarity Matrix Construction

The semantics between different modalities are crucial for efficient and effective SMH approaches. Here, we construct the pairwise similarity matrix with two steps: (1) building up the label matrices $\mathbf{G}_x$/$\mathbf{G}_y$ and normalizing them with each row's l_2-norm to be 1; (2) aligning $\mathbf{S}_{xy}$ to $[-1,+1]^{m \times n}$. More specifically, for the first step, we create the label space with all the shared tags, and then code each sample's labels as a $\{0,1\}^l$ vector, where 1(0) denotes that the image/text has(not) the tag, and l is the dimension of the label space, after which we could obtain the label matrices $\mathbf{G}_x \in \{0,1\}^{m \times l}$ and $\mathbf{G}_y \in \{0,1\}^{n \times l}$; then we normalize each row vector of $\mathbf{G}_x$/$\mathbf{G}_y$ with l_2-norm and achieve the label matrices $\widetilde{\mathbf{G}}_x \in [0,1]^{m \times l}$ and $\widetilde{\mathbf{G}}_y \in [0,1]^{n \times l}$ for images and texts respectively. Regarding the second step, we conduct the following operation:

$$\mathbf{S}_{xy} = 2\widetilde{\mathbf{G}}_x\widetilde{\mathbf{G}}_y^T - \mathbf{1}_m\mathbf{1}_n^T, \tag{3}$$

which makes each element of the semantic matrix $\mathbf{S}_{xy}$ be a real value in the range $[-1,+1]$. Note that, using Eq. (3) with the right side instead of $\mathbf{S}_{xy}$ directly will not only save the storage resources, but also reduce the computational costs in the follow-up learning process. Besides, the reason why we align $\mathbf{S}_{xy}$ to $[-1,+1]^{m \times n}$ is that the label similarity matrix $\mathbf{S}_{xy}$ could be consistent with the binary-codes based similarity matrix $\frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T$, which is the heart of the similarity preserved multi-modal hashing model built in the subsequent section.

3.2 Joint Learning for Hash Codes and Functions

We wish the learned binary codes $\mathbf{B}_x$ and $\mathbf{B}_y$ would well match the semantics $\mathbf{S}_{xy}$, and simultaneously to find out the corresponding hash functions $\mathbf{W}_x$ and $\mathbf{W}_y$. Therefore, a joint learning model is built as below:

$$\begin{aligned} \min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y} & ||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ +\lambda\{&||\mathrm{sgn}(\mathbf{X}\mathbf{W}_x) - \mathbf{B}_x||_F^2 + ||\mathrm{sgn}(\mathbf{Y}\mathbf{W}_y) - \mathbf{B}_y||_F^2\} \\ &+\beta\{||\mathbf{W}_x||_F^2 + ||\mathbf{W}_y||_F^2\} \\ s.t. & \begin{cases} \mathbf{B}_x \in \{-1,+1\}^{m \times q}; \\ \mathbf{B}_y \in \{-1,+1\}^{n \times q}; \\ \mathbf{B}_x^T\mathbf{1}_m = \mathbf{0}_q, \mathbf{B}_x^T\mathbf{B}_x = m\mathbf{I}_q; \\ \mathbf{B}_y^T\mathbf{1}_n = \mathbf{0}_q, \mathbf{B}_y^T\mathbf{B}_y = n\mathbf{I}_q., \end{cases} \end{aligned} \tag{4}$$

where λ is a positive hyper-parameter that balances the importance between semantic matches and hash functions learning, and β is a non-negative smooth factor that avoids overfitting and irreversibility. The discrete constraints are drawn here accompanied by extra balance codings (e.g., $\mathbf{B}_x^T\mathbf{1}_m = \mathbf{0}_q$) and decorrelation of hash bits (e.g., $\mathbf{B}_x^T\mathbf{B}_x = m\mathbf{I}_q$), which would maximize the coding abilities with fixed code lengths [41]. Note that this is distinctive from the current discrete MH methods which only consider the binary-values constraint (e.g., $\mathbf{B}_x \in \{-1,+1\}^{m \times q}$).

Regarding the optimization problem (4), we remove the $\mathrm{sgn}(\cdot)$ function for continuous relaxations, keep the binary constraints, introduce intermediate variables, and finally convert it into:

$$\begin{aligned} \min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y,\mathbf{Z}_x,\mathbf{Z}_y} & ||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ &+\lambda\{||\mathbf{X}\mathbf{W}_x - \mathbf{B}_x||_F^2 + ||\mathbf{Y}\mathbf{W}_y - \mathbf{B}_y||_F^2\} \\ &+\beta\{||\mathbf{W}_x||_F^2 + ||\mathbf{W}_y||_F^2\} \\ s.t. & \begin{cases} \mathbf{B}_x \in \{-1,+1\}^{m \times q}, \mathbf{B}_y \in \{-1,+1\}^{n \times q}; \\ \mathbf{B}_x = \mathbf{Z}_x, \mathbf{B}_y = \mathbf{Z}_y; \\ \mathbf{Z}_x \in \mathbb{R}^{m \times q}, \mathbf{Z}_x^T\mathbf{1}_m = \mathbf{0}_q, \mathbf{Z}_x^T\mathbf{Z}_x = m\mathbf{I}_q; \\ \mathbf{Z}_y \in \mathbb{R}^{n \times q}, \mathbf{Z}_y^T\mathbf{1}_n = \mathbf{0}_q, \mathbf{Z}_y^T\mathbf{Z}_y = n\mathbf{I}_q., \end{cases} \end{aligned} \tag{5}$$

with which the binary codes for training instances can be directly achieved and the hash functions for unseen samples can also be obtained. Here, one should notice that $\mathbf{B}_{x,y}$ and $\mathbf{Z}_{x,y}$ are different in variable scopes, but they are correspondingly equal in values; therefore the constraints in optimization problem (5) essentially still keep the same as the constraints in optimization problem (4). It's worth noting that the intended adaption from the optimization problem (4) to (5) is a special trick in the field of optimization [42], which aims to transform the difficult discrete optimization into a simpler one.

3.3 Overall Objective Function

Firstly, we equivalently replace the pairwise semantic matches $||\mathbf{S}_{xy}-\frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2$ in Eq. (5) with the sum of two terms $\frac{1}{2}||\mathbf{S}_{xy}-\frac{1}{q}\mathbf{Z}_x\mathbf{B}_y^T||_F^2$ and $\frac{1}{2}||\mathbf{S}_{xy}-\frac{1}{q}\mathbf{B}_x\mathbf{Z}_y^T||_F^2$[1], and then relax the equality constraints $\mathbf{B}_x=\mathbf{Z}_x$ and $\mathbf{B}_y=\mathbf{Z}_y$ into the following problem:

$$\begin{aligned}\min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y,\mathbf{Z}_x,\mathbf{Z}_y} &\frac{1}{2}||\mathbf{S}_{xy}-\frac{1}{q}\mathbf{Z}_x\mathbf{B}_y^T||_F^2+\frac{1}{2}||\mathbf{S}_{xy}\\ &-\frac{1}{q}\mathbf{B}_x\mathbf{Z}_y^T||_F^2+\lambda\{||\mathbf{X}\mathbf{W}_x-\mathbf{B}_x||_F^2+||\mathbf{Y}\mathbf{W}_y\\ &-\mathbf{B}_y||_F^2\}+\beta\{||\mathbf{W}_x||_F^2+||\mathbf{W}_y||_F^2\}\\ &+\alpha\{||\mathbf{Z}_x-\mathbf{B}_x||_F^2+||\mathbf{Z}_y-\mathbf{B}_y||_F^2\}\end{aligned} \tag{6}$$

$$s.t.\left\{\begin{array}{l}\mathbf{B}_x\in\{-1,+1\}^{m\times q},\mathbf{B}_y\in\{-1,+1\}^{n\times q};\\ \mathbf{Z}_x\in\mathbb{R}^{m\times q},\mathbf{Z}_x^T\mathbf{1}_m=\mathbf{0}_q,\mathbf{Z}_x^T\mathbf{Z}_x=m\mathbf{I}_q;\\ \mathbf{Z}_y\in\mathbb{R}^{n\times q},\mathbf{Z}_y^T\mathbf{1}_n=\mathbf{0}_q,\mathbf{Z}_y^T\mathbf{Z}_y=n\mathbf{I}_q.,\end{array}\right.$$

where α is a non-negative hyper-parameter to adjust the closeness between $\mathbf{B}_x(\mathbf{B}_y)$ and $\mathbf{Z}_x(\mathbf{Z}_y)$. Since the philosophy of this method is supervised discrete MH with more useful constraints for efficient cross-view retrieval, we call it "Enhanced Discrete Multi-modal Hashing", EDMH for short.

3.4 Out-of-Sample Extension

In practice, we often come across a new image query or a new text query, which could be denoted as $\tilde{\mathbf{x}}\in\mathbb{R}^{d_x}$ or $\tilde{\mathbf{y}}\in\mathbb{R}^{d_y}$ respectively. Their corresponding hash codes are:

$$\mathbf{b}_{\tilde{x}}=\mathbf{H}_x(\tilde{\mathbf{x}})=\mathrm{sgn}(\mathbf{W}_x^T\tilde{\mathbf{x}})\ ; \tag{7}$$

$$\mathbf{b}_{\tilde{y}}=\mathbf{H}_y(\tilde{\mathbf{y}})=\mathrm{sgn}(\mathbf{W}_y^T\tilde{\mathbf{y}})\ . \tag{8}$$

Obviously, the time complexity for coding a new query is quite economical, and the hash functions can be executed in parallel for binarizing large-scale out-of-samples. Since data points from different modalities are efficiently mapped into the shared semantic Hamming space, cross-view retrieval tasks, i.e., image-query-text and text-query-image, could be conducted like a uni-modal retrieval mission.

4 Optimization Algorithm

The optimization problem (6) is not convex in all the six variables together; therefore, we utilize the polular iterative algorithm [15], [18], [26], [31], [42], i.e., alternately optimizing each variable while holding the other five ones fixed, to achieve a local minimum for practical cross-view retrievals.

[1]Note that this step is reasonable because $\mathbf{B}_x=\mathbf{Z}_x$ and $\mathbf{B}_y=\mathbf{Z}_y$.

4.1 $\mathbf{B}_x$-Subproblem

If we optimize $\mathbf{B}_x$ with $\mathbf{B}_y$, $\mathbf{W}_{\{x,y\}}$, and $\mathbf{Z}_{\{x,y\}}$ fixed, then the whole optimization problem is transformed into:

$$\begin{aligned}\min_{\mathbf{B}_x}\ \mathcal{O}=&\frac{1}{2}||\mathbf{S}_{xy}-\frac{1}{q}\mathbf{B}_x\mathbf{Z}_y^T||_F^2+\alpha||\mathbf{Z}_x-\mathbf{B}_x||_F^2\\ &+\lambda||\mathbf{X}\mathbf{W}_x-\mathbf{B}_x||_F^2\\ &s.t.\ \mathbf{B}_x\in\{-1,+1\}^{m\times q}.\end{aligned} \tag{9}$$

Unfolding the objective function (9), we can achieve:

$$\begin{aligned}\mathcal{O}=&\frac{1}{2}tr(\mathbf{S}_{xy}\mathbf{S}_{xy}^T-\frac{2}{q}\mathbf{S}_{xy}\mathbf{Z}_y\mathbf{B}_x^T+\frac{1}{q^2}\mathbf{B}_x\mathbf{Z}_y^T\mathbf{Z}_y\mathbf{B}_x^T)\\ &+\alpha\cdot tr(\mathbf{Z}_x\mathbf{Z}_x^T-2\mathbf{Z}_x\mathbf{B}_x^T+\mathbf{B}_x\mathbf{B}_x^T)\\ &+\lambda\cdot tr(\mathbf{X}\mathbf{W}_x\mathbf{W}_x^T\mathbf{X}^T-2\mathbf{X}\mathbf{W}_x\mathbf{B}_x^T+\mathbf{B}_x\mathbf{B}_x^T)\\ \propto&-\frac{1}{q}tr(\mathbf{S}_{xy}\mathbf{Z}_y\mathbf{B}_x^T)+\frac{1}{2q^2}tr(\mathbf{B}_x n\mathbf{I}_q\mathbf{B}_x^T)\\ &-2\alpha\cdot tr(\mathbf{Z}_x\mathbf{B}_x^T)+\alpha\cdot||\mathbf{B}_x||_F^2\\ &-2\lambda\cdot tr(\mathbf{X}\mathbf{W}_x\mathbf{B}_x^T)+\lambda\cdot||\mathbf{B}_x||_F^2\\ \propto&-tr\{(\frac{1}{q}\mathbf{S}_{xy}\mathbf{Z}_y+2\alpha\mathbf{Z}_x+2\lambda\mathbf{X}\mathbf{W}_x)\mathbf{B}_x^T\},\end{aligned}$$

based on which the optimization (9) is equivalent to:

$$\begin{aligned}\max_{\mathbf{B}_x}\ &tr\{(\frac{1}{q}\mathbf{S}_{xy}\mathbf{Z}_y+2\alpha\mathbf{Z}_x+2\lambda\mathbf{X}\mathbf{W}_x)\mathbf{B}_x^T\}\\ &s.t.\ \mathbf{B}_x\in\{-1,+1\}^{m\times q}.\end{aligned} \tag{10}$$

Although the problem (10) is a discrete optimization problem, we could directly work it out as follows:

$$\mathbf{B}_x=\mathrm{sgn}(\frac{1}{q}\mathbf{S}_{xy}\mathbf{Z}_y+2\alpha\mathbf{Z}_x+2\lambda\mathbf{X}\mathbf{W}_x). \tag{11}$$

4.2 $\mathbf{B}_y$-Subproblem

It's easy to find that the optimization of $\mathbf{B}_y$ is almost the same with $\mathbf{B}_x$-Subproblem; therefore, the optimal solution of $\mathbf{B}_y$-Subproblem could be written into:

$$\mathbf{B}_y=\mathrm{sgn}(\frac{1}{q}\mathbf{S}_{xy}^T\mathbf{Z}_x+2\alpha\mathbf{Z}_y+2\lambda\mathbf{Y}\mathbf{W}_y). \tag{12}$$

4.3 $\mathbf{W}_x$-Subproblem

With $\mathbf{B}_{\{x,y\}}$, $\mathbf{W}_y$, and $\mathbf{Z}_{\{x,y\}}$ fixed, the optimization w.r.t. $\mathbf{W}_x$ is simplified as:

$$\min_{\mathbf{W}_x}\ \mathcal{O}=\lambda||\mathbf{X}\mathbf{W}_x-\mathbf{B}_x||_F^2+\beta||\mathbf{W}_x||_F^2. \tag{13}$$

Unfolding the objective function (13) and setting the derivative of $\mathbf{W}_x$ to zero matrix, we could arrive at:

$$\frac{\partial\mathcal{O}}{\partial\mathbf{W}_x}=2\lambda(\mathbf{X}^T\mathbf{X}+\frac{\beta}{\lambda}\mathbf{I}_{d_x})\mathbf{W}_x-2\lambda\mathbf{X}^T\mathbf{B}_x=\mathbf{O}; \tag{14}$$

then the optimal $\mathbf{W}_x$ for the problem (13) is:

$$\mathbf{W}_x=(\mathbf{X}^T\mathbf{X}+\frac{\beta}{\lambda}\mathbf{I}_{d_x})^{-1}\mathbf{X}^T\mathbf{B}_x. \tag{15}$$

4.4 $\mathbf{W}_y$-Subproblem

Similar as the $\mathbf{W}_x$-Subproblem, the optimum of $\mathbf{W}_y$-Subproblem is exhibited as:

$$\mathbf{W}_y=(\mathbf{Y}^T\mathbf{Y}+\frac{\beta}{\lambda}\mathbf{I}_{d_y})^{-1}\mathbf{Y}^T\mathbf{B}_y. \tag{16}$$

Algorithm 1: EDMH

Input: Data matrices $\mathbf{X}$ and $\mathbf{Y}$; Label matrices $\mathbf{G}_x$ and $\mathbf{G}_y$; Hyper-parameters α, β, and λ; Length of binary codes q; Maximum iterations $maxIter$.

Output: $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{Z}_x$, and $\mathbf{Z}_y$.

randomly initialize $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{Z}_x$ with each element be a real value between -1 and $+1$; and $\mathbf{Z}_y = \mathbf{Z}_x$;

$\mathbf{B}_x = \mathrm{sgn}(\mathbf{Z}_x)$, $\mathbf{B}_y = \mathrm{sgn}(\mathbf{Z}_y)$;

for $index$=1:$maxIter$ **do**

  update $\mathbf{B}_x$ according to Eq. (11);

  update $\mathbf{B}_y$ according to Eq. (12);

  update $\mathbf{W}_x$ according to Eq. (15);

  update $\mathbf{W}_y$ according to Eq. (16);

  update $\mathbf{Z}_x$ according to Eq. (19);

  update $\mathbf{Z}_y$ using the similar solution as Eq. (19) according to Lemma 1;

end

return $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{W}_x$, $\mathbf{W}_y$, $\mathbf{Z}_x$, and $\mathbf{Z}_y$.

4.5 $\mathbf{Z}_x$-Subproblem

When fixing the variables $\mathbf{B}_{\{x,y\}}$, $\mathbf{W}_{\{x,y\}}$ and $\mathbf{Z}_y$, and optimizing $\mathbf{Z}_x$, the optimization problem is reduced to:

$$\min_{\mathbf{Z}_x} \mathcal{O} = \frac{1}{2}||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{Z}_x\mathbf{B}_y^T||_F^2 + \alpha||\mathbf{Z}_x - \mathbf{B}_x||_F^2 \qquad (17)$$
$$s.t.\ \mathbf{Z}_x \in \mathbb{R}^{m\times q}, \mathbf{Z}_x^T\mathbf{1}_m = \mathbf{0}_q, \mathbf{Z}_x^T\mathbf{Z}_x = m\mathbf{I}_q,$$

which could be further simplified as:

$$\max_{\mathbf{Z}_x} tr(\mathbf{E}_x\mathbf{Z}_x^T) \qquad (18)$$
$$s.t.\ \mathbf{Z}_x \in \mathbb{R}^{m\times q}, \mathbf{Z}_x^T\mathbf{1}_m = \mathbf{0}_q, \mathbf{Z}_x^T\mathbf{Z}_x = m\mathbf{I}_q,$$

where $\mathbf{E}_x = \frac{1}{q}\mathbf{S}_{xy}\mathbf{B}_y + 2\alpha\mathbf{B}_x$. Set the centering matrix $\mathbf{J} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ and then do singular value decomposition (SVD) of $\mathbf{JE}_x$ as $\mathbf{JE}_x = \mathbf{U}\Sigma\mathbf{V}^T = \sum_{k=1}^{r'}\sigma_k\mathbf{u}_k\mathbf{v}_k^T$, where $r' \leq q$ is the rank of $\mathbf{JE}_x$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r'}$ are the positive singular values, and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots \mathbf{u}_{r'}]$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_{r'}]$. Next, by employing the Gram-Schmidt process, we can obtain matrices $\bar{\mathbf{U}} \in \mathbb{R}^{m\times(q-r')}$ and $\bar{\mathbf{V}} \in \mathbb{R}^{q\times(q-r')}$ such that $\bar{\mathbf{U}}^T\bar{\mathbf{U}} = \mathbf{I}_{q-r'}$, $[\mathbf{U}, \mathbf{1}_m]^T\bar{\mathbf{U}} = \mathbf{O}$ and $\bar{\mathbf{V}}^T\bar{\mathbf{V}} = \mathbf{I}_{q-r'}$, $\mathbf{V}^T\bar{\mathbf{V}} = \mathbf{O}$[2]. To solve the optimization (18), we could borrow the following lemma:

Lemma 1. $\mathbf{Z}_x = \sqrt{m}[\mathbf{U}, \bar{\mathbf{U}}][\mathbf{V}, \bar{\mathbf{V}}]^T$ *is the optimal solution to the maximization problem (18).*

Proof. Please refer to [43]. □

Therefore, we can re-write the final optimal solution as:

$$\mathbf{Z}_x = \sqrt{m}[\mathbf{U}, \bar{\mathbf{U}}][\mathbf{V}, \bar{\mathbf{V}}]^T\ . \qquad (19)$$

4.6 $\mathbf{Z}_y$-Subproblem

Imitating the above $\mathbf{Z}_x$-Subproblem, we could also find the equivalent optimization as below:

$$\max_{\mathbf{Z}_y} tr(\mathbf{E}_y\mathbf{Z}_y^T) \qquad (20)$$
$$s.t.\ \mathbf{Z}_y \in \mathbb{R}^{n\times q}, \mathbf{Z}_y^T\mathbf{1}_n = \mathbf{0}_q, \mathbf{Z}_y^T\mathbf{Z}_y = n\mathbf{I}_q,$$

[2] if $r' = q$, then $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ will be nothing.

where $\mathbf{E}_y = \frac{1}{q}\mathbf{S}_{xy}^T\mathbf{B}_x + 2\alpha\mathbf{B}_y$. Obviously, the optimization problem (20) is in the same form with optimization problem (18); thus, we could utilize the result in Lemma 1 for $\mathbf{Z}_y$-Subproblem.

Based on the above six subproblems, we could conclude the iterative learning process in Algorithm 1. Since each subproblem has an efficient closed-form solution, the whole algorithm is quite fast and its time complexity is liner to the size of dataset whose specific details are provided in the following.

4.7 Complexity Analysis

Although there are six variables to be optimized in Algorithm 1, we just need to concentrate on three ones w.r.t. the computational complexities because of the model parameters' symmetry. Let's take $\mathbf{B}_x$, $\mathbf{W}_x$ and $\mathbf{Z}_x$ into considerations.

First, the time complexity of Eq. (11) for solving $\mathbf{B}_x$ is O($(m+n)lq+md_xq$). Here one should notice "$\mathbf{S}_{xy} = 2\mathbf{G}_x\mathbf{G}_y^T - \mathbf{1}_m\mathbf{1}_n^T$", which reduces the complexity of O(mnq) to O($(m+n)lq$).

Second, in terms of the Eq. (15), it will cost O($md_x^2 + d_x^3 + mqd_x + qd_x^2$). Typically, d_x, q will be much less than m; then this step's time complexity will be linear to the number of samples.

Last, let's take an analysis for the Eq. (19). The main time-consuming steps would be the SVD of $\mathbf{JE}_x = \mathbf{E}_x - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T\mathbf{E}_x$ and its time complexity is O($(m+n)lq + mq^2$).

Obviously, the above analysis reveals that the time complexity for each iteration in Algorithm 1 is linear to the size of datasets. In addition, the updating times for convergence are usually within 10 iterations (please refer to Fig. 3), which means the liner time complexity of the whole EDMH algorithm.

5 More Modalities

The proposed EDMH could be extended straightforwadly to the scenario of more than two modalities. In practice, there are two commom strategies to accompolish this which are described as follows. (1) The first is to repeat the EDMH algorithm for each combination of two modalities, e.g., there are $\binom{4}{2} = \frac{4\times3}{2\times1} = 6$ combinations for four modalities, such as (image, text, video, audio) [1], [44]. (2) The second is to build a new joint model based on the principle of EDMH, which takes the pairwise similarities between any two modalities into considerations. Details of this strategy are specified in the following paragraph.

Suppose that there are data instances $\mathbf{X}$ consisting of t ($t \geq 2$) modalities' matrices, denoted by $\mathbf{X}_i$ ($i = 1, 2\cdots, t$), and the matrices $\mathbf{S}_{ij}$ represents the pairwise similarities between the i-th and the j-th modality. Then the extension of EDMH can be written into:

$$\begin{aligned}\min_{\{\mathbf{B}_i,\mathbf{W}_i,\mathbf{Z}_i\}_{i=1,2\cdots,t}} \ & \frac{1}{2}\sum_{1\leq i<j\leq t}^{t}||\mathbf{S}_{ij} - \frac{1}{q}\mathbf{Z}_i\mathbf{B}_j^T||_F^2 \\ & +\frac{1}{2}\sum_{1\leq i<j\leq t}^{t}||\mathbf{S}_{ij} - \frac{1}{q}\mathbf{B}_i\mathbf{Z}_j^T||_F^2 \\ & +\lambda\sum_{i=1}^{t}||\mathbf{X}_i\mathbf{W}_i - \mathbf{B}_i||_F^2 + \beta\sum_{i=1}^{t}||\mathbf{W}_i||_F^2 \\ & +\alpha\sum_{i=1}^{t}||\mathbf{Z}_i - \mathbf{B}_i||_F^2 \\ s.t. & \begin{cases}\mathbf{B}_i \in \{-1,+1\}^{n_i\times q}; \\ \mathbf{Z}_i \in \mathbb{R}^{n_i\times q}; \\ \mathbf{Z}_i^T\mathbf{1}_{n_i} = \mathbf{0}_q, \mathbf{Z}_i^T\mathbf{Z}_i = n_i\mathbf{I}_q.,\end{cases}\end{aligned} \qquad (21)$$

Table 2
Statistics of three benchmark datasets.

	Wiki	MIRFlickr	NUS-WIDE
#Labels	10	24	10
#Training Set	2,173	15,902	184,711
#Testing Set	693	836	1,866

where $\mathbf{B}_i$, $\mathbf{W}_i$, and $\mathbf{Z}_i$ corresponds to the i-th modality's binary codes, hash function and intermediate variable, and α, β and λ are non-negative hyper-parameters to balance the contributions of different items. Here $n = n_i$ denotes the number of training instances in dataset $\mathbf{X}$. In light of the solutions for $\mathbf{B}_i$, $\mathbf{W}_i$ and $\mathbf{Z}_i$, it is not difficult to find that they could be straightforward borrowed from the above optimization in EDMH. Since the essences of EDMH and its extension are the same, we would mainly testify the high performance of the bi-modal version (i.e., EDMH) in the sequel for simplicity.

6 Experiments

We have conducted extensive experiments to evaluate EDMH's effectiveness and efficiency, using a commodity PC with Intel®Core$^{\text{TM}}$ i7-4790 CPU@3.60GHz 4-Cores and 32GB RAM.

6.1 Datasets

Three public benchmarks, i.e., a single-labeled Wiki [45] dataset and two multiple-labeled datasets MIRFLickr [46] and NUS-WIDE [47] with different scales, are adopted for evaluating the multi-modal retrieval performance.

Wiki originates from Wikipedia's featured articles, and it consists of $2,866$ image-text pairs annotated with 10 semantic labels. For each pair, the image is coded as a 128-dimensional SIFT feature vector and the text is represented as a 10-dimensional topic vector generated by Latent Dirichlet Allocation (LDA) [48]. The dataset is divided into two parts: $2,173$ image-text pairs and 693 image-text pairs for training and testing sets respectively.

MIRFlickr is crawled from Flickr with $25,000$ instances, each being an image with some associated textual tags. In our experiments, we only keep those image-text pairs that contain textual tags appearing at least 20 times, and then achieve a $16,738$-scale collection. For each instance, the image is vectorized with 150-dimensional edge histograms and the text is represented by a 500-dimensional vector derived from PCA on the binary textual tags. 5% of the dataset are randomly selected as the testing set, and the others come to the training set.

NUS-WIDE is a real-world web database originally containing $269,648$ instances, with each being an image and associated textual tags. In accordance with the protocol in [23], we also choose those instances that cover the top 10 most frequent semantic concepts and finally obtain $186,577$ image-text pairs. Regarding such instances, the images are expressed by 500-dimensional bag-of-visual-word features and texts are coded as 1000-dimensional vectors of the most frequent tags. Here we take 1% of the dataset as the testing set and the remaining as the training set.

For all the datasets, the key statistics are summarized in Table 2. Note that two instances sharing at least one tag are considered to be relevant in the retrieval experiments.

6.2 Evaluation

Considering the existing comparable approaches whose codes are publicly available, we select some representative and state-of-the-art MH methods as baselines: CMFH[3] [15], [49], LSSH[4] [16], STMH[5] [17], FSH[6] [14], CRE[7] [18], CMSSH[8] [20], SCM[9] [22], SePH[10] [23], GSPH[11] [24], DCH[12] [26], DLFH/KDLFH[13] [31]. With respect to our proposed EDMH method, the code will be published online. Since all the methods are in Matlab codes, we could further record the time cost for each approach and compare their fastness. Note that the first five are unsupervised MH baselines, and the rest are supervised MH approaches.

The proposed MH method is evaluated by different measurements, i.e., Precision, Recall, Mean Average Precision (MAP), Precision-Recall curves and the time cost, which are widely used in the field of hashing such as in Refs. [45], [50], [51], [52], [53], [54]. Precision/Recall@topN measure the precision and recall at fixed levels of retrieved results, and they don't take into account the rank order within the topN retrieved items; MAP and Precision-Recall curves are both to evaluate the overall performance of the retrieval systems; and the time expenditure is recorded to assess how fast the MH methods will be.

6.3 Settings

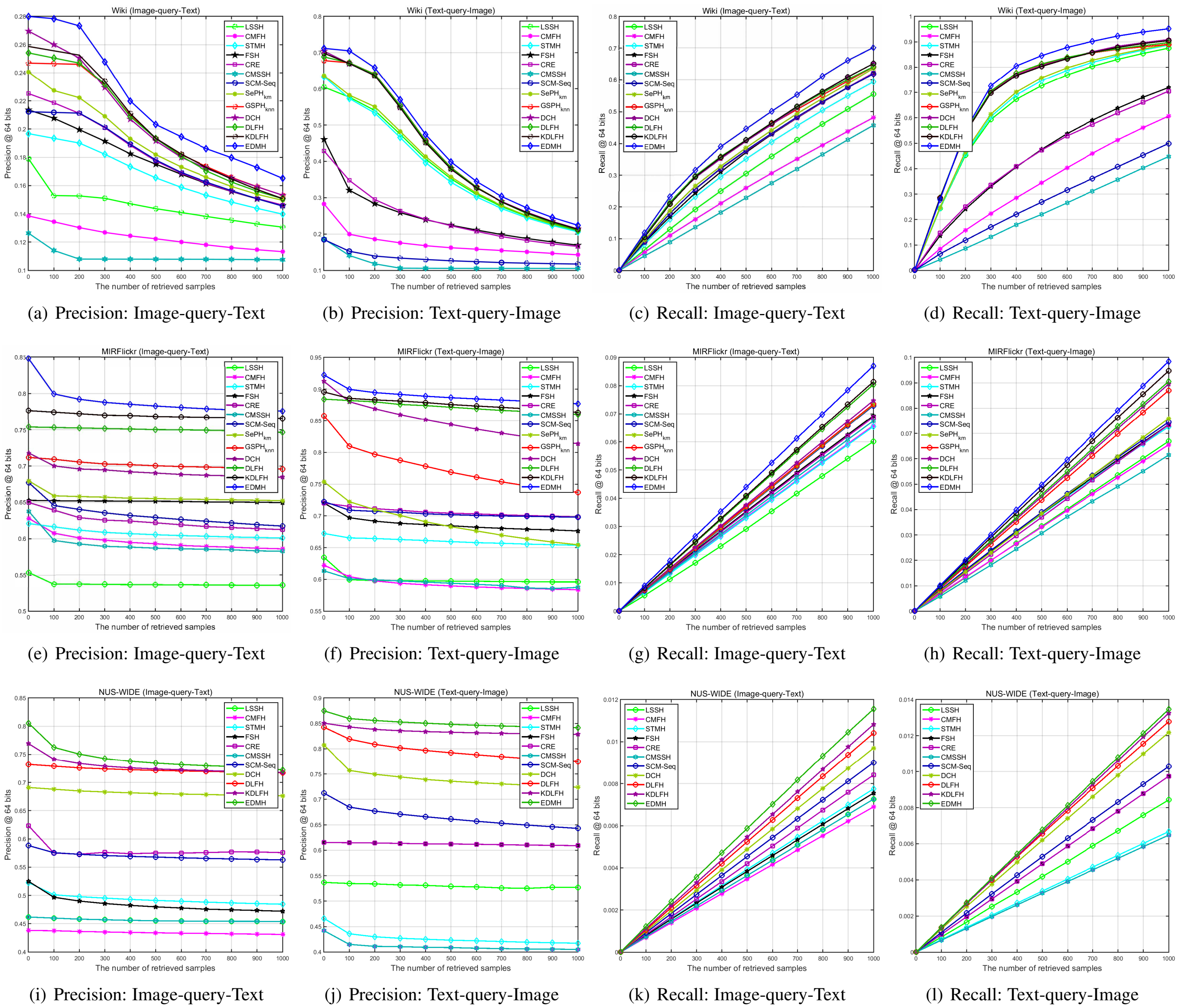
To guarantee a fair comparison, we first make the inputs (i.e., the data and label matrices) for all the competing methods identical. Then in terms of the baseline methods, we conduct initializations according to the corresponding papers and tune them on different datasets for the most competitive performances.

With respect to our EDMH method, the $maxIter$ is configured as 10 because the EDMH algorithm could converge fast (please see Fig. 3). Regarding the other hyper-parameters α, β and λ, we empirically settle a fixed group with (α, β, λ)=(0.1, 1.0, 10), and then vary each one ranging from 0, 10^{-9} to 10^{9} and choose the best while keeping the other two unchanged; finally, we arrive at (α=0.1, β=0.1, λ=1.0), which would yield most competitive retrieval performances on all datasets.

6.4 Results

Fig. 1 exhibits the Precision and Recall for the topN returned results with 64 bits[14], Fig. 2 plots the Precision-Recall curves with 64 bits[14], and Table 3 displays the MAP values with various code lengths on the three benchmark datasets. Clearly, we can see that whether it's Precision/Recall@topN, Precision-Recall curves or MAP, EDMH consistently outperforms all the baseline methods for all the various settings, which testifies its effectiveness in cross-view retrieval tasks. Particularly, even compared with most competitive KDLFH, EDMH still exhibits clear advantages. It's worth mentioning that KDLFH is a nonlinear/kernel learning

[3] http://ise.thss.tsinghua.edu.cn/MIG/code_data_cmfh.zip
[4] http://ise.thss.tsinghua.edu.cn/MIG/LSSH_code.rar
[5] The matlab code is kindly provided by the author.
[6] https://github.com/LynnHongLiu/FSH
[7] We implement the algorithm with Matlab.
[8] http://www.cs.technion.ac.il/∼mbron/publications_conference.html
[9] http://cs.nju.edu.cn/lwj/code/SCMHash_Release.zip
[10] https://sites.google.com/site/linzijia72/
[11] https://github.com/devraj89/Generalized-Semantic-Preserving-Hashing-for-N-Label-Cross-Modal-Retrieval
[12] http://cfm.uestc.edu.cn/∼fshen/pub.html
[13] https://github.com/jiangqy/DLFH-TIP2019
[14] The results with other number of bits are similar to those with 64 bits.



(a) Precision: Image-query-Text (b) Precision: Text-query-Image (c) Recall: Image-query-Text (d) Recall: Text-query-Image

(e) Precision: Image-query-Text (f) Precision: Text-query-Image (g) Recall: Image-query-Text (h) Recall: Text-query-Image

(i) Precision: Image-query-Text (j) Precision: Text-query-Image (k) Recall: Image-query-Text (l) Recall: Text-query-Image

Figure 1. Precision and Recall curves of cross-view retrieval tasks on three datasets with 64 bits (best viewed in color).

method, while our EDMH is just a linear approach. Such superior performance can attribute to EDMH's capabilities to well preserve cross-modalities' semantics in the Hamming space as well as the joint learning for hash functions.

Besides, we also record the time costs of all the methods and report them in Table 4. Obviously, the fastest approach is SCM-Seq which mainly benefits from its sequential bit-wise optimizations; however, its retrieval performance is very limited. SePH$_{km}$ and GSPH$_{knn}$ are well-performed MH methods (Table 3), but they are so resource-consuming that they can't proceed successfully on the large NUS-WIDE dataset. Regarding our EDMH, its time cost is much less than that of most baseline approaches; even compared with the unsupervised competitors (e.g., LSSH, STMH, FSH and CRE), EDMH still performs faster especially on larger databases. This merit probably owes the efficient similarity matrix constructions and the carefully designed algorithm.

To sum up, the high retrieval performance and the economical time expenditures endow the EDMH model with more capabilities to handle large-scale cross-modal retrieval.

6.5 Convergence Analysis

The updating rules for minimizing the objective function of EDMH are essentially iterative, and it's easy to verify that these rules will converge to a local minimum. Here, we would mainly investigate how fast EDMH can converge.

Fig. 3 displays the convergence curves of EDMH on all the three datasets with 32/64-bit code lengths. For this figure, the y-axis is the normalized objective function value[15] and the x-axis denotes the iteration number. We can clearly see that the designed Algorithm 1 converges quite fast, usually within 10 iterations, which probably benefits from the efficient closed-form solutions of the subproblems.

6.6 Parameter Sensitivity Analysis

Further experiments are conducted to analyze the influence of parameters (α, β and λ) on the cross-modal retrieval performance.

[15] Each iteration's loss is divided by the first iteration's loss.

Table 3
The MAP results of all methods on three datasets with various hash code lengths. Note that "—" represents that the approaches can't be executed successfully on large training set (NUS-WIDE) because of their high space and computational complexities.

Tasks/Methods		Wiki				MIRFlickr				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Image Query v.s. Text Database	LSSH	0.2130	0.2201	0.2227	0.2241	0.5784	0.5835	0.5866	0.5876	0.4115	0.4115	0.4080	0.4012
	CMFH	0.1960	0.2061	0.2119	0.2184	0.5867	0.5865	0.5843	0.5842	0.3830	0.3857	0.3871	0.3877
	STMH	0.2165	0.2334	0.2532	0.2617	0.5740	0.5738	0.5754	0.5632	0.4276	0.4524	0.4512	0.4504
	FSH	0.2344	0.2475	0.2540	0.2654	0.6254	0.6267	0.6328	0.6307	0.4907	0.5104	0.5090	0.5161
	CRE	0.2474	0.2544	0.2668	0.2672	0.6128	0.6183	0.6238	0.6288	0.5007	0.5191	0.5231	0.5266
	CMSSH	0.1742	0.1735	0.1608	0.1509	0.5973	0.5846	0.5863	0.5839	0.4226	0.4191	0.4055	0.4042
	SCM-Seq	0.2341	0.2410	0.2453	0.2566	0.6280	0.6345	0.6385	0.6490	0.5125	0.5418	0.5513	0.5476
	SePH$_{km}$	0.2735	0.2853	0.3070	0.3135	0.6331	0.6349	0.6368	0.6389	—	—	—	—
	GSPH$_{knn}$	0.2743	0.2960	0.2967	0.3055	0.6713	0.6792	0.6853	0.6837	—	—	—	—
	DCH	0.3350	0.3523	0.3701	0.3737	0.6545	0.6842	0.6989	0.6965	0.5730	0.5916	0.5993	0.6175
	DLFH	0.2991	0.3075	0.3529	0.3672	0.6858	0.7012	0.7119	0.7337	0.6025	0.6457	0.6576	0.6588
	KDLFH	0.3187	0.3575	0.3665	0.3686	0.7048	0.7194	0.7388	0.7414	0.6281	0.6479	0.6621	0.6683
	EDMH	**0.3420**	**0.3684**	**0.3756**	**0.3781**	**0.7393**	**0.7444**	**0.7578**	**0.7606**	**0.6462**	**0.6690**	**0.6780**	**0.6811**
Text Query v.s. Image Database	LSSH	0.5008	0.5243	0.5311	0.5386	0.5882	0.5940	0.5968	0.5959	0.4419	0.4443	0.4291	0.4141
	CMFH	0.4816	0.5120	0.5235	0.5427	0.5965	0.5949	0.5944	0.5915	0.4164	0.4213	0.4150	0.4051
	STMH	0.5253	0.5400	0.5451	0.5581	0.5945	0.5952	0.5987	0.5980	0.4341	0.4417	0.4260	0.4102
	FSH	0.4994	0.5199	0.5136	0.5694	0.6167	0.6155	0.6212	0.6194	0.4670	0.4837	0.4839	0.4906
	CRE	0.4933	0.5148	0.5153	0.5219	0.6072	0.6182	0.6191	0.6296	0.4719	0.4756	0.4783	0.4818
	CMSSH	0.1629	0.1670	0.1638	0.1576	0.5945	0.5937	0.5833	0.5834	0.3944	0.3872	0.3732	0.3675
	SCM-Seq	0.2257	0.2459	0.2490	0.2524	0.6176	0.6234	0.6285	0.6369	0.4777	0.5000	0.5102	0.5068
	SePH$_{km}$	0.6431	0.6512	0.6692	0.6693	0.6623	0.6620	0.6658	0.6679	—	—	—	—
	GSPH$_{knn}$	0.6512	0.6640	0.6675	0.6758	0.7216	0.7366	0.7424	0.7427	—	—	—	—
	DCH	0.6996	0.7088	0.7018	0.7065	0.7311	0.7608	0.7919	0.8136	0.7187	0.7314	0.7343	0.7590
	DLFH	0.6589	0.6738	0.6852	0.6896	0.7799	0.8170	0.8189	0.8262	0.7580	0.7764	0.7861	0.7874
	KDLFH	0.6825	0.7053	0.7065	0.7101	0.8096	0.8223	0.8287	0.8345	0.7693	0.7982	0.8061	0.8074
	EDMH	**0.7078**	**0.7195**	**0.7211**	**0.7118**	**0.8190**	**0.8298**	**0.8392**	**0.8413**	**0.7862**	**0.8001**	**0.8093**	**0.8139**

Table 4
Time cost (in seconds) of the training stage on benchmark datasets for different approaches with different hash code lengths.

Methods/ Time Cost (Seconds)	Wiki				MIRFlickr				NUS-WIDE			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
LSSH	295.78	329.28	377.69	436.67	241.02	262.86	266.36	270.28	378.79	391.19	412.46	523.62
CMFH	0.09	0.14	0.19	0.36	4.22	4.90	6.24	9.50	468.57	489.78	545.16	665.06
STMH	1.15	1.81	3.89	8.74	14.77	26.42	52.76	137.11	3842.81	3996.06	4111.73	4861.19
FSH	1.17	1.23	2.76	3.68	72.91	73.52	74.90	80.42	4735.30	4827.79	4859.79	4914.77
CRE	0.76	1.06	1.47	2.43	34.04	44.51	57.63	89.63	459.04	488.73	551.62	733.42
CMSSH	65.76	131.16	260.98	517.15	57.61	97.81	177.36	336.54	118.67	215.78	427.93	1285.32
SCM-Seq	0.23	0.34	0.53	0.66	0.64	0.56	1.14	2.97	19.42	36.77	52.58	96.57
SePH$_{km}$	54.84	87.53	155.15	285.64	635.73	676.27	773.82	947.97	—	—	—	—
GSPH$_{knn}$	260.95	674.01	1034.34	2701.94	6359.66	18418.07	24541.24	46207.58	—	—	—	—
DCH	0.37	0.73	3.35	8.47	6.17	13.71	36.44	130.29	155.40	175.04	344.29	1063.88
DLFH	1.29	1.60	11.90	52.22	5.59	24.12	95.21	389.01	80.16	279.93	1048.69	3291.28
KDLFH	122.51	255.93	377.59	596.51	413.68	815.43	1648.17	2405.33	5317.67	7965.39	13105.91	17124.71
EDMH	0.55	0.88	2.11	3.02	5.56	7.10	8.51	16.58	127.38	145.14	187.67	284.18

In particular, the MAP curves of EDMH on different datasets with 64 bits are drawn in Fig. 4[16]. From this figure, we could observe that EDMH generates good retrieval performances with a large wide range of values regarding parameters α and β; while it's a little different in terms of parameter λ. Even so, it's probably showing the trend (from Fig. 4(g) to Fig. 4(i)) that with larger-scale datasets, EDMH is not that sensitive to parameter λ over a wider range. Finally, we arrive at a group of configurations, i.e., $(\alpha, \beta, \lambda) = (0.1, 0.1, 1.0)$, for competitive performances.

Noticeably, from Fig. 4(a) to Fig. 4(c), the MAP values are almost the same when $\alpha = 0$ and $0 < \alpha < 0.1$, which indicates that this regularizer almost makes no effect. Recall that when we build the overall objective function in EDMH, the following replacement is conducted:

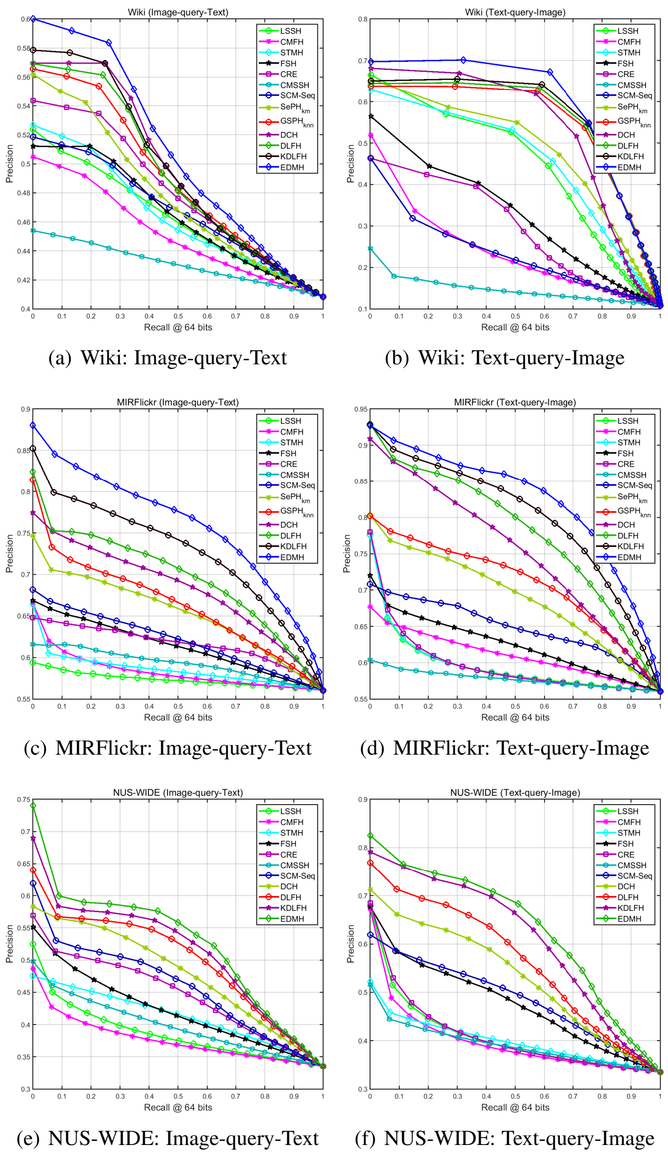
$$\begin{aligned} &||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ =&\frac{1}{2}||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{Z}_x\mathbf{B}_y^T||_F^2 + \frac{1}{2}||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{Z}_y^T||_F^2, \end{aligned} \tag{22}$$

which essentially implies that $\mathbf{B}_x = \mathbf{Z}_x$ and $\mathbf{B}_y = \mathbf{Z}_y$, i.e., it equivalently contains the regularizer:

$$||\mathbf{Z}_x - \mathbf{B}_x||_F^2 + ||\mathbf{Z}_y - \mathbf{B}_y||_F^2. \tag{23}$$

Besides, compared with Eq. (22), Eq. (23) just accounts for a very small proportion in the whole objection function; thus Eq. (23) could be seen as being absorbed by Eq. (22) when $\alpha < 0.1$, which

[16] The MAP curves with other code lengths are similar to Fig. 4.



(a) Wiki: Image-query-Text (b) Wiki: Text-query-Image

(c) MIRFlickr: Image-query-Text (d) MIRFlickr: Text-query-Image

(e) NUS-WIDE: Image-query-Text (f) NUS-WIDE: Text-query-Image

Figure 2. Precision-Recall curves on different datasets with 64 bits.

probably explains why there is no obvious differences between $\alpha = 0$ and $0 < \alpha < 0.1$ in terms of EDMH's performance.

In regard to the parameter β, it functions as a smooth factor in case of overfitting and irreversibility; therefore it's quite common to see that when $\beta = 0$, EDMH stills performs almost the same with $\beta < 0.1$ (from Fig. 4(d) to Fig. 4(f)) in complex real-world datasets (i.e., in the selected datasets, there doesn't exist the phenomenon of overfitting and irreversibility).

To conclude, we here set $\alpha = 0.1$ and $\beta = 0.1$ instead of zeros for generalizations to other datasets.

6.7 Ablation Study

To investigate the contributions of such "$\mathbf{B}^T\mathbf{1} = \mathbf{0}, \mathbf{B}^T\mathbf{B} = n\mathbf{I}_q$" constraints in discrete MH, we deliberately remove one or two from EDMH and get three weaker models called EDMH$_W$, EDMH$_{W+B}$ and EDMH$_{W+D}$, corresponding to the optimization problem (24), (25) and (26) respectively as below:

$$\begin{aligned} \min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y,\mathbf{Z}_x,\mathbf{Z}_y} & ||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ & +\lambda\{||\mathbf{X}\mathbf{W}_x - \mathbf{B}_x||_F^2 + ||\mathbf{Y}\mathbf{W}_y - \mathbf{B}_y||_F^2\} \\ & +\beta\{||\mathbf{W}_x||_F^2 + ||\mathbf{W}_y||_F^2\} \\ s.t.\ & \mathbf{B}_x \in \{-1,+1\}^{m\times q}, \mathbf{B}_y \in \{-1,+1\}^{n\times q}; \end{aligned} \tag{24}$$

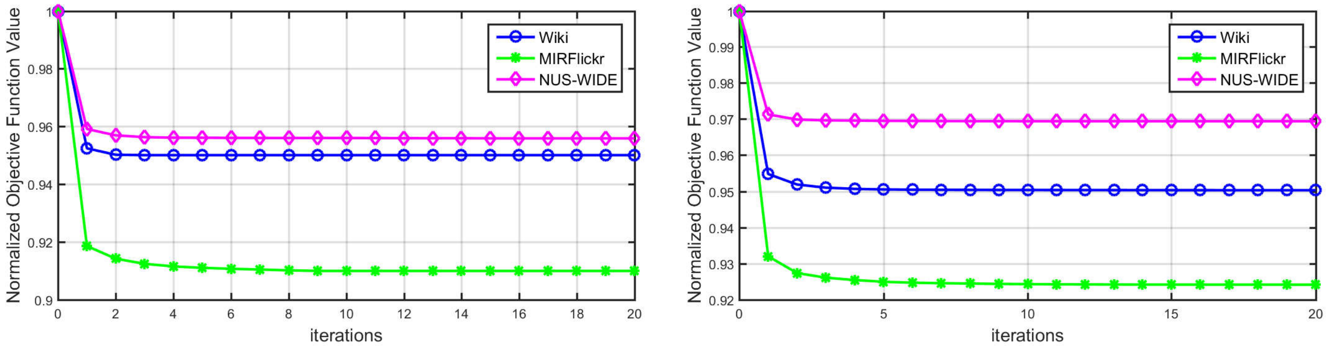
$$\begin{aligned} \min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y,\mathbf{Z}_x,\mathbf{Z}_y} & ||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ & +\lambda\{||\mathbf{X}\mathbf{W}_x - \mathbf{B}_x||_F^2 + ||\mathbf{Y}\mathbf{W}_y - \mathbf{B}_y||_F^2\} \\ & +\beta\{||\mathbf{W}_x||_F^2 + ||\mathbf{W}_y||_F^2\} \\ s.t.\ & \begin{cases} \mathbf{B}_x \in \{-1,+1\}^{m\times q}, \mathbf{B}_y \in \{-1,+1\}^{n\times q}; \\ \mathbf{B}_x^T\mathbf{1}_m = \mathbf{0}_q, \mathbf{B}_y^T\mathbf{1}_n = \mathbf{0}_q; \end{cases} \end{aligned} \tag{25}$$

$$\begin{aligned} \min_{\mathbf{B}_x,\mathbf{B}_y,\mathbf{W}_x,\mathbf{W}_y,\mathbf{Z}_x,\mathbf{Z}_y} & ||\mathbf{S}_{xy} - \frac{1}{q}\mathbf{B}_x\mathbf{B}_y^T||_F^2 \\ & +\lambda\{||\mathbf{X}\mathbf{W}_x - \mathbf{B}_x||_F^2 + ||\mathbf{Y}\mathbf{W}_y - \mathbf{B}_y||_F^2\} \\ & +\beta\{||\mathbf{W}_x||_F^2 + ||\mathbf{W}_y||_F^2\} \\ s.t.\ & \begin{cases} \mathbf{B}_x \in \{-1,+1\}^{m\times q}, \mathbf{B}_y \in \{-1,+1\}^{n\times q}; \\ \mathbf{B}_x^T\mathbf{B}_x = m\mathbf{I}_q, \mathbf{B}_y^T\mathbf{B}_y = n\mathbf{I}_q. \end{cases} \end{aligned} \tag{26}$$

Regarding the discrete optimization techniques to address the above discrete hashing problems, there are two popular paradigms, namely, the *discrete cyclic coordinate descent* (DCC) method [56] and the *dciscrete proximal linearized minimization* (DPLM) [55]. However, in this paper, we employed the DPLM with the following reasons: (1) DCC adopts the bit-wise optimization strategy which could only solves the binary constrained problem (i.e., EDMH$_W$), and at the same time is usually time-consuming when the hash code length is long (e.g., 128 bits) [57], [58]. (2) In contrast, DPLM is a fast optimization method for general binary code learning, which could tackle the above three adapted models (EDMH$_W$, EDMH$_{W+B}$ and EDMH$_{W+D}$) in a unified form, and is faster than DCC as discussed in Ref. [55].

To ensure a fair competition between EDMH and its variants, we solve EDMH, EDMH$_W$, EDMH$_{W+B}$ and EDMH$_{W+D}$ with DPLM, trying best to tune the involved parameters according to the proposals in Ref. [55] for the best performance. Specifically, the model parameters are configured with ($\beta = 0.01$, $\lambda = 10$), and the DLPM algorithm's parameters are set the same with those in Ref. [55]. In what follows, the MAP results of EDMH with/without "$\mathbf{B}^T\mathbf{1} = \mathbf{0}$" or "$\mathbf{B}^T\mathbf{B} = n\mathbf{I}_q$" constraints on three various datasets are collected and displayed in Table 5.

From Table 5, we could draw several critical discoveries: (1) Compared with EDMH$_W$, EDMH$_{W+B}$ and EDMH$_{W+D}$ both yield much higher MAP scores on all the selected datasets with different hash code lengths, which validates the "balance codings" and the "decorrelation of hash bits" can both make contributions to multi-modal hashing for better cross-view retrievals. (2) Besides, "EDMH+DPLM" further shows superior performance to EDMH$_{W+B}$ and EDMH$_{W+D}$, which testifies the great benefits of the integrated constraints "$\mathbf{B}^T\mathbf{1} = \mathbf{0}$, $\mathbf{B}^T\mathbf{B} = n\mathbf{I}_q$". (3) What's more, take "EDMH+DPLM" and "EDMH" into account, it's easy to conclude that the proposed Algorithm 1 owns more advantages than DPLM in solving the EDMH model.



(a) 32bits (b) 64bits

Figure 3. Convergence curves of EDMH on various datasets with 32 and 64 bits (Notice: other bit-settings hold similar results).

Table 5
The MAP results of EDMH with/without "$\mathbf{B}^T\mathbf{1}=\mathbf{0}$" or "$\mathbf{B}^T\mathbf{B}=n\mathbf{I}_q$" constraints on various benchmark datasets. Note that "EDMH+DPLM" represents that the EDMH model is solved with the "Discrete Proximal Linearized Minimization (DPLM)" algorithm [55], which is distinctive from our Algorithm 1.

Tasks/Methods		Wiki				MIRFlickr				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Image Query v.s. Text Database	EDMH$_W$	0.2905	0.3240	0.3355	0.3395	0.6449	0.6524	0.6549	0.6585	0.5859	0.5972	0.6029	0.6162
	EDMH$_{W+B}$	0.3123	0.3327	0.3461	0.3598	0.6573	0.6676	0.6716	0.6786	0.6020	0.6117	0.6133	0.6249
	EDMH$_{W+D}$	0.3214	0.3361	0.3519	0.3567	0.6549	0.6747	0.6762	0.6764	0.6023	0.6138	0.6188	0.6229
	EDMH+DPLM	0.3278	0.3441	0.3603	0.3651	0.6659	0.6798	0.6825	0.6928	0.6174	0.6266	0.6320	0.6447
	EDMH	**0.3420**	**0.3684**	**0.3756**	**0.3781**	**0.7393**	**0.7444**	**0.7578**	**0.7606**	**0.6462**	**0.6690**	**0.6780**	**0.6811**
Text Query v.s. Image Database	EDMH$_W$	0.6326	0.6543	0.6577	0.6580	0.6844	0.6961	0.7219	0.7279	0.6432	0.6535	0.6547	0.6561
	EDMH$_{W+B}$	0.6485	0.6564	0.6667	0.6694	0.6920	0.7182	0.7441	0.7593	0.6657	0.6687	0.6718	0.6764
	EDMH$_{W+D}$	0.6508	0.6614	0.6637	0.6686	0.7007	0.7265	0.7532	0.7553	0.6551	0.6710	0.6785	0.6787
	EDMH+DPLM	0.6625	0.6745	0.6759	0.6777	0.7174	0.7346	0.7606	0.7645	0.6686	0.6877	0.7091	0.7203
	EDMH	**0.7078**	**0.7195**	**0.7211**	**0.7118**	**0.8190**	**0.8298**	**0.8392**	**0.8413**	**0.7862**	**0.8001**	**0.8093**	**0.8139**

Table 6
Time cost (in seconds) of the training stage on benchmark datasets for EDMH with different optimization algorithms.

Datasets/Methods		16 bits	32 bits	64 bits	128 bits
Wiki	EDMH+DPLM	1.52	3.711	8.10	17.72
	EDMH	0.55	0.88	2.11	3.02
MIRFlickr	EDMH+DPLM	14.423	22.32	47.43	79.22
	EDMH	5.56	7.10	8.51	16.58
NUS-WIDE	EDMH+DPLM	155.53	189.65	292.75	495.69
	EDMH	127.38	145.14	187.67	284.18

By the way, we have also recorded the time cost of the training stage on three benchmark datasets for EDMH and "EDMH+DPLM" in Table 6. Clearly, no matter how long the hash code is set, EDMH runs much faster than "EDMH+DPLM", which exhibits that our algorithm is quite fast efficient.

Overall, more constraints, indeed make the EDMH model more complex and challenging, but meanwhile they make it more effective and efficient, unleashing its potential for scalable cross-view retrieval.

7 Unpaired Multi-modal Data

The above has investigated the EDMH's performance on common scenarios with one-to-one correspondence between images and texts (e.g., Wiki [45], Flickr [46] and NUS-WIDE [47]). In what follows, we further explored its cross-model retrieval behaviors on more general application scenerios, i.e., with mixed paired and unpaired image-text couples.

7.1 Re-constructed Datasets

To testify EDMH's abilities on unpaired scenarios, we should re-construct the benchmark datasets. In fact, we could continue to maintain the testing set of Section 6, and just remove some images or/and texts from the training set of Section 6. Fig. 5 illustrates how to reshape the training datasets. Specifically, the first subfigure in Fig. 5 completely contains the one-to-one image-text pairs (i.e., $paired : unpaired = 100\% : 0$) as a reference, based on which two different unpaired scenarios are built. The second subfigure in Fig. 5 wipes off 10% samples from both images and texts, i.e., $paired : unpaired = 80\% : 20\%$. Similarly, the third subfigure in Fig. 5 forms a case with $paired : unpaired = 40\% : 60\%$. Here, we call the latter two cases "**unpaired scenarios**", which highly simulate the more general practical retrieval systems with both paired and unpaired image-text couples.

Hence, based on the above criterias, six unpaired training sets could be re-constructed in two groups, i.e., $\{$Wiki (8:2), MIRFlickr (8:2), NUS-WIDE (8:2)$\}$ and $\{$Wiki (4:6), MIRFlickr (4:6), NUS-WIDE (4:6)$\}$.

7.2 Settings

Section 6 has employed more than 10 baseline approaches to hold a cross-retrieval competition on the "**paired**" scenario. However, when talkig about the more general "**unpaired**" scenario, only

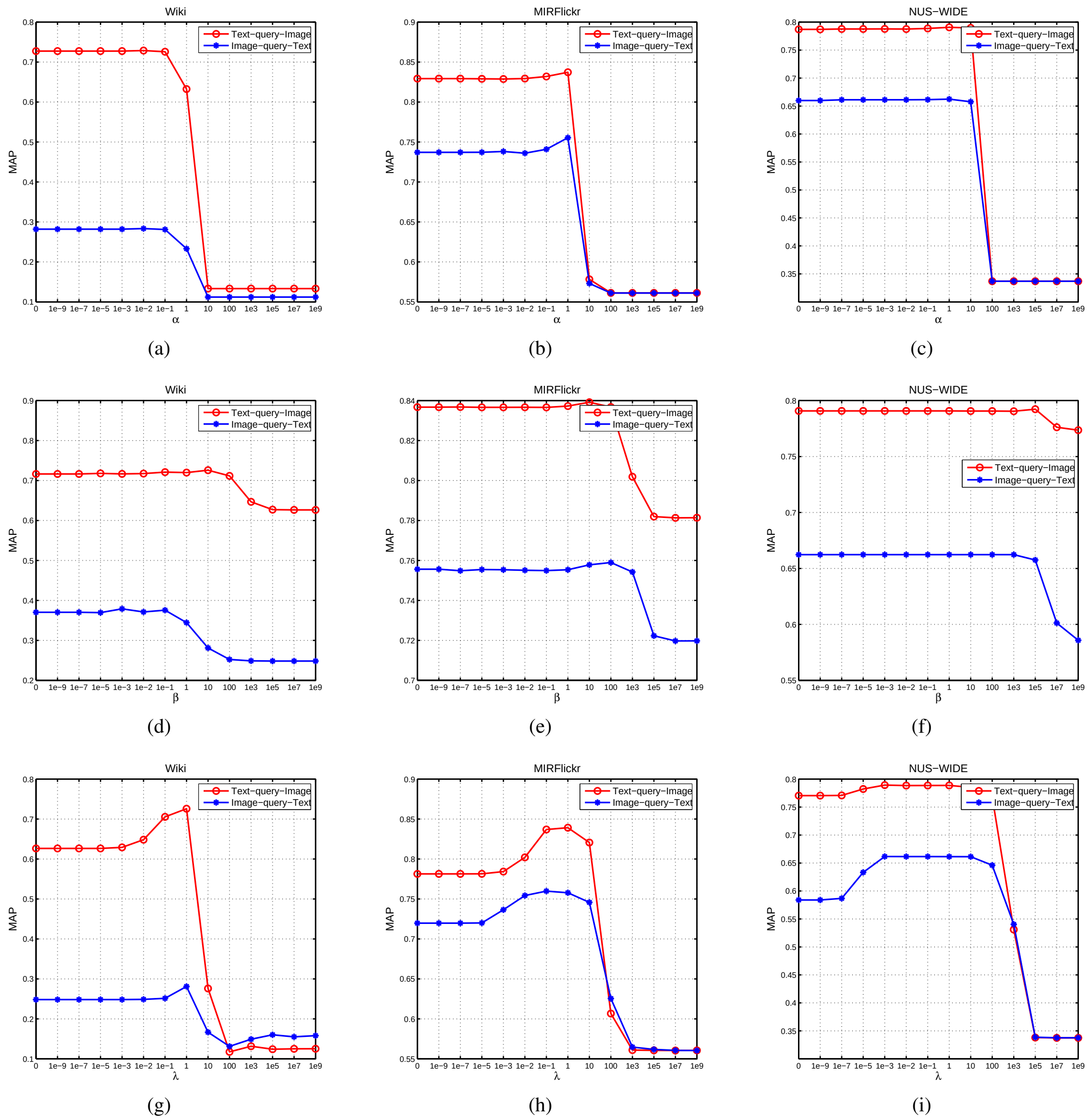


Figure 4. Parameter sensitivity analysis of (α, β, and λ) on different datasets with 64 bits.

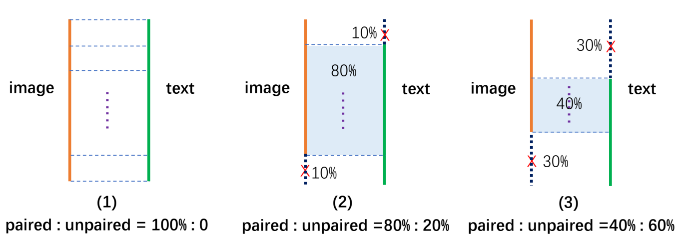


Figure 5. Datasets re-constructions: mixed paired and unpaired image-text couples with different ratios.

two, i.e., CMSSH [20] and GSPH [24], [25], are left. Note that, to the best of our knowledge, GSPH is most competitive in cross-modal retrievals with unpaired settings. Thus, we mainly compare our EDMH with them (CMSSH/GSPH$_{knn}$) on the re-constructed datasets. With respect to other setups, such as the codes, configurations and measurements, they are the same with those in Section 6.

7.3 Results

Table 7 collects the MAP values of selected methods with various code lengths on the three ***unpaired*** benchmark datasets. Note that the best results are in bold, and clearly we can find that EDMH defeats the other two competitors almost in all the different settings (even on the very minor cases, EDMH is quite close to the best GSPH$_{knn}$), which overall validates its high-performace in cross-view retrieval tasks.

Besides, the competitors' time cost (in seconds) of the training stage on the three ***unpaired*** benchmark datasets with different hash code lengths are also recorded in Table 8. Undoubtedly, EDMH exhibits evident advantages over other methods. Particularly, on the 10k+-scale dataset MIRFlickr, GSPH$_{knn}$ took as high as several hours in sharp contrast to about 10 seconds in EDMH. In addition, on a larger-scale NUS-WIDE dataset, the current most competitive method GSPH$_{knn}$ failed because of its high space and computing expenditures. Nevertheless, EDMH runs successfully in just two or three minutes, which actually reveals our proposed method's great potentials in real-world retrieval systems.

Table 7
The MAP results of selected methods on three ***Unpaired*** datasets with various hash code lengths. Note that "—" represents that the approaches can't be executed successfully on large training set (NUS-WIDE) because of their high space and computational complexities.

Tasks/Methods		Wiki				MIRFlickr				NUS-WIDE			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
Image Query v.s. Text Database	CMSSH (8:2)	0.1614	0.1615	0.1676	0.1685	0.5539	0.5688	0.5697	0.5704	0.4016	0.4122	0.4144	0.4172
	GSPH_{knn} (8:2)	0.2544	0.3008	0.3045	0.3165	**0.6562**	0.6577	0.6767	0.6877	—	—	—	—
	EDMH (8:2)	**0.3260**	**0.3387**	**0.3440**	**0.3690**	0.6537	**0.6865**	**0.6916**	**0.7183**	**0.6376**	**0.6399**	**0.6451**	**0.6604**
	CMSSH (4:6)	0.1517	0.1592	0.1680	0.1694	0.5985	0.5935	0.5947	0.5985	0.3878	0.4052	0.4131	0.4176
	GSPH_{knn} (4:6)	0.2517	0.3034	**0.3194**	0.3164	0.6512	0.6829	0.6861	0.6906	—	—	—	—
	EDMH (4:6)	**0.3018**	**0.3087**	0.3124	**0.3210**	**0.6591**	**0.7094**	**0.7425**	**0.7458**	**0.5946**	**0.6117**	**0.6358**	**0.6373**
Text Query v.s. Image Database	CMSSH (8:2)	0.1553	0.1508	0.1666	0.1637	0.5666	0.5698	0.5710	0.5786	0.3749	0.3787	0.3795	0.3848
	GSPH_{knn} (8:2)	0.6404	0.6626	0.6639	0.6743	0.6979	**0.7307**	0.7428	0.7481	—	—	—	—
	EDMH (8:2)	**0.6960**	**0.7105**	**0.7130**	**0.7200**	**0.7050**	0.7221	**0.7442**	**0.8046**	**0.7365**	**0.7391**	**0.7412**	**0.7521**
	CMSSH (4:6)	0.1515	0.1531	0.1535	0.1633	0.5782	0.5758	0.5813	0.5866	0.3614	0.3728	0.3767	0.3825
	GSPH_{knn} (4:6)	0.6352	0.6723	0.6738	0.6879	0.6974	0.7212	0.7556	0.7603	—	—	—	—
	EDMH (4:6)	**0.6492**	**0.6792**	**0.6856**	**0.6962**	**0.7089**	**0.8018**	**0.8149**	**0.8234**	**0.6487**	**0.7003**	**0.7392**	**0.7541**

Table 8
Time cost (in seconds) of the training stage on three ***Unpaired*** benchmark datasets for different approaches with different hash code lengths.

Methods/ Time Cost (Seconds)	Wiki				MIRFlickr				NUS-WIDE			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CMSSH (8:2)	55.69	117.44	241.02	412.87	44.42	87.58	168.73	293.51	102.07	186.18	396.89	1186.35
GSPH_{knn} (8:2)	234.24	576.02	954.86	2315.37	5590.41	15772.07	21293.00	39966.74	—	—	—	—
EDMH (8:2)	0.49	0.54	1.56	2.77	4.41	6.38	7.37	14.31	116.23	124.99	172.69	267.87
CMSSH (4:6)	46.03	86.04	171.89	286.26	34.88	76.23	122.85	204.73	74.73	141.80	295.43	718.34
GSPH_{knn} (4:6)	202.44	493.19	799.08	2012.61	4733.78	13619.97	17954.80	32910.61	—	—	—	—
EDMH (4:6)	0.20	0.25	0.44	2.21	3.63	5.41	6.70	12.88	89.27	105.51	145.28	183.28

In short, the EDMH's talent in cross-view retrievals is further unveiled in more general scenarios with mixed paired and unparied image-text couples.

8 Conclusions

This paper mainly tries to tackle the discrete MH with more constraints (i.e., balance codings and decorrelation), and then puts forward a novel pairwise semantics preserved MH method in the joint learning framework. Regarding the proposed complex and challenging EDMH model, two auxiliary variables are introduced to simplify the optimization, which triggers an effective and efficient solution. Noticeably, EDMH has linear time complexity and thus is very suitable for large-scale cross-view retrieval. Experiments on three image-text collections (with both paired and unpaired settings) show that EDMH can achieve better retrieval performances than many state-of-the-art methods.

For future work, we would like to examine multi-modal hashing on bigger multi-modal datasets, with more and diverse class labels. In partciular, it would be interesting to see how different multi-modal hashing methods work on the recent *open long-tailed datasets*[17] [59] where we must deal with significant data imbalance and probably need to incorporate few-/zero-shot learning techniques. Furthermore, tapping into the full power of deep learning is certainly attractive for multi-modal hashing [40], epscially because different types of data could be processed end-to-end by a unified neural network architecture. As mentioned before, the major difficulty for a widespread usage of deep learning in multi-modal hashing has been the lack of massive labelled data. Utilizing adversarial learning [60], [61] or self-supervised learning [62], [63] to generate pseudo-labels looks a very promising way to overcome this obstacle and further improve the performance of multi-modal hashing.

[17] https://github.com/zhmiao/OpenLongTailRecognition-OLTR

Acknowledgments

This work is supported in part by the China Postdoctoral Science Foundation (grant No. 8206300295) and the National Key Research and Development Program of China (grant No. 2017YFB1400200). Besides, we also thank the Network Information Center of Beihang University (BUAA) for providing high-performance servers.

References

[1] H. Li, J. Zhu, C. Ma, J. Zhang, and C. Zong, "Read, watch, listen, and summarize: Multi-modal summarization for asynchronous text, image, audio and video," *IEEE Trans. Knowl. Data Eng.*, pp. 996–1009, 2019. 1, 5

[2] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, "Deep visual-semantic hashing for cross-modal retrieval," in *KDD*, 2016, pp. 1445–1454. 1

[3] K. Barnard and D. A. Forsyth, "Learning the semantics of words and pictures," in *ICCV*, 2001, pp. 408–415. 1

[4] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu, "Automatic multi-media cross-modal correlation discovery," in *KDD*, 2004, pp. 653–658. 1

[5] N. Chen, J. Zhu, F. Sun, and E. P. Xing, "Large-margin predictive latent subspace learning for multiview data analysis," *IEEE Trans. on Pattern Anal. Mach. Intell.*, pp. 2365–2378, 2012. 1

[6] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *ACM Multimedia*, 2011, pp. 423–432. 1

[7] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang, "Sparse multi-modal hashing," *IEEE Trans. on Multimedia*, pp. 427–439, 2014. 1

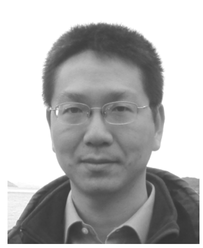
[8] P. Wu, S. C. H. Hoi, P. Zhao, C. Miao, and Z. Liu, "Online multi-modal distance metric learning with application to image retrieval," *IEEE Trans. Knowl. Data Eng.*, pp. 454–467, 2016. 1
[9] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang, "Quantized correlation hashing for fast cross-modal search," in *IJCAI*, 2015, pp. 3946–3952. 1
[10] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *VLDB*, 1999, pp. 518–529. 1
[11] A. Dasgupta, R. Kumar, and T. Sarlós, "Fast locality-sensitive hashing," in *KDD*, 2011, pp. 1073–1081. 1
[12] Q. Huang, J. Feng, Q. Fang, and W. Ng, "Two efficient hashing schemes for high-dimensional furthest neighbor search," *IEEE Trans. Knowl. Data Eng.*, pp. 2772–2785, 2017. 1
[13] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *SIGMOD*, 2013, pp. 785–796. 1
[14] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *CVPR*, 2017, pp. 6345–6353. 1, 6
[15] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *CVPR*, 2014, pp. 2083–2090. 1, 2, 4, 6
[16] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *SIGIR*, 2014, pp. 415–424. 1, 6
[17] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multimodal hashing for cross-media retrieval," in *IJCAI*, 2015, pp. 3890–3896. 1, 6
[18] M. Hu, Y. Yang, F. Shen, N. Xie, R. Hong, and H. T. Shen, "Collective reconstructive embeddings for cross-modal hashing," *IEEE Trans. Image Processing*, pp. 2770–2784, 2019. 1, 4, 6
[19] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, pp. 30–37, 2004. 2
[20] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *CVPR*, 2010, pp. 3594–3601. 2, 6, 11
[21] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *IJCAI*, 2011, pp. 1360–1365. 2
[22] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *AAAI*, 2014, pp. 2177–2183. 2, 6
[23] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *CVPR*, 2015, pp. 3864–3872. 2, 6
[24] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for n-label cross-modal retrieval," in *CVPR*, 2017, pp. 2633–2641. 2, 6, 11
[25] ——, "Generalized semantic preserving hashing for cross-modal retrieval," *IEEE Trans. Image Processing*, pp. 102–112, 2019. 2, 11
[26] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Processing*, pp. 2494–2507, 2017. 2, 4, 6
[27] X. Luo, X.-Y. Yin, L. Nie, X. Song, Y. Wang, and X.-S. Xu, "SDMCH: Supervised discrete manifold-embedded cross-modal hashing," in *IJCAI*, 2018, pp. 2518–2524. 2
[28] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, pp. 2323–2326, 2000. 2
[29] C.-X. Li, Z.-D. Chen, P.-F. Zhang, X. Luo, L. Nie, W. Zhang, and X.-S. Xu, "SCRATCH: A scalable discrete matrix factorization hashing for cross-modal retrieval," in *ACM Multimedia*, 2018, pp. 1–9. 2
[30] X. Luo, P.-F. Zhang, Y. Wu, Z.-D. Chen, H.-J. Huang, and X.-S. Xu, "Asymmetric discrete cross-modal hashing," in *ICMR*, 2018, pp. 204–212. 2
[31] Q.-Y. Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Processing*, pp. 3490–3501, 2019. 2, 4, 6
[32] Y. Luo, Y. Yang, F. Shen, Z. Huang, P. Zhou, and H. T. Shen, "Robust discrete code modeling for supervised hashing," *Pattern Recognition*, pp. 128–135, 2018. 2
[33] H. T. Shen, L. Liu, Y. Yang, X. Xu, Z. Huang, F. Shen, and R. Hong, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowledge and Data Engineering*, 2020. 2
[34] Z.-D. Chen, W.-J. Yu, C.-X. Li, L. Nie, and X.-S. Xu, "Dual deep neural networks cross-modal hashing," in *AAAI*, 2018, pp. 274–281. 2
[35] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *AAAI*, 2017, pp. 1618–1625. 2
[36] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *CVPR*, 2017, pp. 3270–3278. 2
[37] X. Li, D. Hu, and F. Nie, "Deep binary reconstruction for cross-modal hashing," in *ACM Multimedia*, 2017, pp. 1398–1406. 2
[38] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Trans. Image Processing*, pp. 3893–3903, 2018. 2
[39] L. Zhen, P. Hu, X. Wang, and D. Peng, "Deep supervised cross-modal retrieval," in *CVPR*, 2019, pp. 10 394–10 403. 2
[40] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 3034–3044, 2018. 2, 12
[41] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS*, 2008, pp. 1753–1760. 3
[42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, pp. 1–122, 2011. 4
[43] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *NIPS*, 2014, pp. 3419–3427. 5
[44] X. He, Y. Peng, and L. Xie, "A new benchmark and approach for fine-grained cross-media retrieval," in *ACM Multimedia*, 2019, pp. 1740–1748. 5
[45] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *ACM Multimedia*, 2010, pp. 251–260. 6, 10
[46] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *MIR*, 2008, pp. 39–43. 6, 10
[47] 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 *CIVR*, 2009. 6, 10
[48] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," in *NIPS*, 2001, pp. 601–608. 6
[49] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans. on Image Processing*, pp. 5427–5440, 2016. 6
[50] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling Internet images, tags, and their semantics," *International Journal of Computer Vision*, pp. 210–233, 2014. 6
[51] A. Sharma, A. Kumar, H. D. III, and D. W. Jacobs, "Generalized multiview analysis: A discriminative latent space," in *CVPR*, 2012, pp. 2160–2167. 6
[52] N. Rasiwasia, D. Mahajan, V. Mahadevan, and G. Aggarwal, "Cluster canonical correlation analysis," in *AISTATS*, 2014, pp. 823–831. 6
[53] K. Wang, R. He, W. Wang, L. Wang, and T. Tan, "Learning coupled feature spaces for cross-modal matching," in *ICCV*, 2013, pp. 2088–2095. 6
[54] C. Kang, S. Xiang, S. Liao, C. Xu, and C. Pan, "Learning consistent feature representation for cross-modal multimedia retrieval," *IEEE Trans. on Multimedia*, pp. 370–381, 2015. 6
[55] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, "A fast optimization method for general binary code learning," *IEEE Trans. Image Processing*, pp. 5610–5621, 2016. 9, 10
[56] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *CVPR*, 2015, pp. 37–45. 9
[57] W.-C. Kang, W.-J. Li, and Z.-H. Zhou, "Column sampling based discrete supervised hashing," in *AAAI*, 2016, pp. 1230–1236. 9
[58] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 490–496, 2018. 9
[59] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *CVPR*, 2019, pp. 2537–2546. 12
[60] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *ACM Multimedia*, 2017, pp. 154–162. 12
[61] J. Song, T. He, L. Gao, X. Xu, A. Hanjalic, and H. T. Shen, "Binary generative adversarial networks for image retrieval," in *AAAI*, 2018, pp. 394–401. 12
[62] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, "Self-supervised adversarial hashing networks for cross-modal retrieval," in *CVPR*, 2018, pp. 4242–4251. 12
[63] G. Wu, J. Han, Z. Lin, G. Ding, B. Zhang, and Q. Ni, "Joint image-text hashing for fast large-scale cross-media retrieval using self-supervised deep learning," *IEEE Trans. Industrial Electronics*, pp. 9868–9877, 2019. 12

Yong Chen received his Ph.D. in Computer Science and Engineering from Beihang University (BUAA), Beijing, China, in 2019. He is currently working as a "Boya" Postdoc in the Key Lab of Machine Perception, School of EECS, Peking University, Beijing, China. He has been funded as a visiting Ph.D. student at Birkbeck and UCL from January 2018 to January 2019. His research interests include machine learning, data mining and numerical optimization. For more information, please refer to https://scholar.google.com/citations?user=bakW4s4AAAAJ&hl=zh-CN.

Hui Zhang received the M.S. and Ph.D. degrees in computer science from Beihang University, Beijing, China, in 1994 and 2009, respectively. He is a Professor and also the Deputy Director at State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University. He had been working in the University of Chicago and Argonne National Laboratory, Chicago, IL, USA, from 2007 to 2008 as a Guest Researcher. His main research interests include e-science archives management, data mining, and information retrieval.

Zhibao Tian received B.Sc. degree from computer science and technology, China University of Geosciences, Beijing, PR China, in 2017. Now he is a master student in the State Key Lab of Software Development Environment, School of Computer Science and Engineering, Beihang University, Beijing, China. He received gold award at the 40th ACM/ICPC International College Student Programming Contest, Asia Changchun Station during his undergraduate period. His main research interests include machine learning, data mining, and big data, especially information retrieval and recommender system for large-scale S&T resources.

Jun Wang is Chair Professor, Computer Science, University College London, and Founding Director of MSc Web Science and Big Data Analytics. Prof. Jun Wang's main research interests are in the areas of AI and intelligent systems, including (multiagent) reinforcement learning, deep generative models, and their diverse applications on information retrieval, recommender systems and personalization, data mining, smart cities, bot planning, computational advertising etc. His team won the first global real-time bidding algorithm contest with 80+ participants worldwide. Jun has published over 100 research papers and is a winner of multiple "Best Paper" awards. He was a recipient of the Beyond Search - Semantic Computing and Internet Economics award by Microsoft Research and also received Yahoo! FREP Faculty award. He has served as an Area Chair in ACM CIKM and ACM SIGIR. His recent service includes co-chair of Artificial Intelligence, Semantics, and Dialog in ACM SIGIR 2018. For more information, please refer to http://www0.cs.ucl.ac.uk/staff/Jun.Wang/bio.html.



Dell Zhang is a Reader in Computer Science at Birkbeck, University of London (UoL), a Senior Member of ACM, a Senior Member of IEEE, and a Fellow of RSS. He is currently on leave from Birkbeck and working for Blue Prism AI Labs. He got his PhD from Southeast University, Nanjing, China, and then worked as a Research Fellow at the Singapore-MIT Alliance (SMA) until he moved to the UK in 2005. His research interests include Natural Language Processing, Information Retrieval, and Machine Learning. He has received multiple best paper awards, and won prizes from several data science competitions. For more information, please refer to http://www.dcs.bbk.ac.uk/~dell/.

Xuelong Li (M'02-SM'07-F'12) is a full professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, China.