

# Label Consistent Matrix Factorization Hashing for Large-Scale Cross-Modal Similarity Search

Di Wang<sup>ID</sup>, Xinbo Gao<sup>ID</sup>, Senior Member, IEEE, Xiumei Wang<sup>ID</sup>, and Lihuo He<sup>ID</sup>, Member, IEEE

**Abstract**—Multimodal hashing has attracted much interest for cross-modal similarity search on large-scale multimedia data sets because of its efficiency and effectiveness. Recently, supervised multimodal hashing, which tries to preserve the semantic information obtained from the labels of training data, has received considerable attention for its higher search accuracy compared with unsupervised multimodal hashing. Although these algorithms are promising, they are mainly designed to preserve pairwise similarities. When semantic labels of training data are given, the algorithms often transform the labels into pairwise similarities, which gives rise to the following problems: (1) constructing pairwise similarity matrix requires enormous storage space and a large amount of calculation, making these methods unscalable to large-scale data sets; (2) transforming labels into pairwise similarities loses the category information of the training data. Therefore, these methods do not enable the hash codes to preserve the discriminative information reflected by labels and, hence, the retrieval accuracies of these methods are affected. To address these challenges, this paper introduces a simple yet effective supervised multimodal hashing method, called label consistent matrix factorization hashing (LCMFH), which focuses on directly utilizing semantic labels to guide the hashing learning procedure. Considering that relevant data from different modalities have semantic correlations, LCMFH transforms heterogeneous data into latent semantic spaces in which multimodal data from the same category share the same representation. Therefore, hash codes quantified by the obtained representations are consistent with the semantic labels of the original data and, thus, can have more discriminative power for cross-modal similarity search tasks. Thorough experiments on standard databases show that the proposed algorithm outperforms several state-of-the-art methods.

**Index Terms**—Hashing, multimodal, supervised, similarity search, cross-modal

## 1 INTRODUCTION

WITH the explosive growth of multimedia data, performing efficient and accurate similarity searches on large-scale data sets has become a challenging issue [1], [2]. To address large-scale similarity search problems, various hashing-based methods that, map similar data points in the original feature space to adjacent binary codes in a low-dimensional Hamming space have been proposed for their remarkable efficiency gains and storage reductions. Early endeavors in hashing are unimodal methods [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. They are focused on learning compact hash codes for a single type of features from unimodal data. However, with the rapid development of the Internet and social networks, a tremendous amount of multimedia data has been generated. It is common for multimedia data with the same semantic to exist in different modalities. For example, an image in Flickr is often associated with related text descriptions, or a microblog in Facebook consists of paired texts and

images. Therefore, it is desirable to design hashing methods over multimodal data sets.

To facilitate cross-modal retrieval, many multimodal hashing methods have been proposed in recent years [15], [16]. According to whether supervised information is used, multimodal hashing methods can be divided into two categories: unsupervised methods and supervised methods [17]. Unsupervised methods usually explore correlations from heterogeneous data and preserve the obtained correlations for binary codes. There are mainly two types of unsupervised multimodal hashing methods. Graph-based methods construct similarity graphs to preserve correlations for the hash codes. However, they suffer from high training complexity for computing the similarity graphs [18], [19], [20]. Matrix decomposition-based methods avoid the large-scale graph construction process. They seek latent semantic spaces to find correlations in multimodal data [21], [22], [23], [24]. However, these methods cannot utilize supervised information such as semantic labels to further improve the search accuracy.

Supervised multimodal hashing methods, which try to preserve the semantic correlations obtained from the semantic labels of training points, have exhibited higher accuracies than unsupervised multimodal hashing methods. Therefore, they have received increasingly more attention in recent years. The first attempt at supervised multimodal hashing was the cross-modality similarity-sensitive hashing (CMSHH) [25]. In this approach, the hashing process is expressed as a binary classification problem with positive and negative pairs and can be efficiently learned from the perspective of boosting.

- D. Wang is with the School of Computer Science and Technology, Xidian University, No.2, South Taibai Road, Xi'an 710071, China.  
E-mail: wangdi.wandy@gmail.com.
- X. Gao, X. Wang, and L. He are with the Video and Image Processing System (VIPS) Lab, School of Electronic Engineering, Xidian University, No.2, South Taibai Road, Xi'an 710071, China. E-mail: xbgaomail.xidian.edu.cn, wangxm@xidian.edu.cn, lihuo.he@gmail.com.

Manuscript received 1 Apr. 2016; revised 14 May 2018; accepted 23 July 2018. Date of publication 30 July 2018; date of current version 12 Sept. 2019.

(Corresponding author: Xinbo Gao.)

Recommended for acceptance by K. Crammer.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2018.2861000

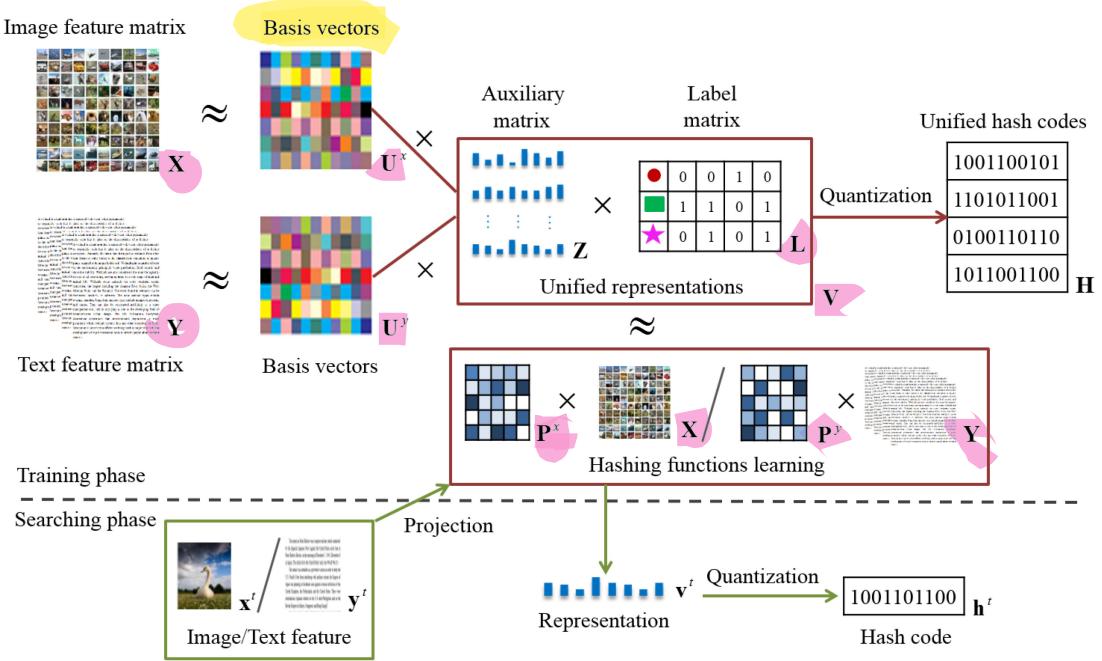


Fig. 1. Framework of label consistent matrix factorization hashing.

However, CMSSH preserves only the inter-modal correlations and ignores the intra-modal correlations. To preserve the intra-modal and inter-modal similarities at the same time, Kumar et al. extended the traditional spectral hashing to the multimodal setting and proposed cross-view hashing (CVH). CVH minimizes the weighted average distances between data pairs by solving a generalized eigenvalue problem [26]. Nevertheless, the eigenvalue decomposition process decreases the mapping quality substantially when increasing the bit number, because most of the variance is contained in the top few eigenvectors. Multi latent binary embedding (MLBE) employs a generative probabilistic model to learn the binary latent factors hidden in data and regards the binary latent factors as hash codes [27]. However, hash codes generated by MLBE do not require independence between different hash bits, and may result in highly redundant hash bits. Semantics-preserving hashing (SEPH) approximates a pairwise similarity matrix with to-be-learnt hash codes by minimizing the Kullback-Leibler divergence (KLD) between them [28]. The aforementioned supervised multimodal hashing methods are mainly designed to preserve pairwise similarities. When semantic labels instead of pairwise similarities are available, the preceding methods often transform labels into pairwise similarities which lead to high training time complexity. Semantic correlation maximization (SCM) avoids this type of transformation process by seamlessly integrating semantic labels into the hashing learning procedure for large-scale data modeling [29]. It constructs the pairwise semantic similarity by the cosine similarity between the semantic label vectors and then approximates the pairwise similarity matrix with the to-be-learnt hash codes.

While few attempts have been made towards supervised multimodal hashing, these methods often share the common objective of preserving pairwise similarities for hash codes. However, such a strategy often causes the following problems. First, most existing methods suffer from high time complexity when semantic labels instead of pairwise

similarities are available [25], [26], [27], [28]. Specifically, the time complexity for constructing a pairwise similarity graph is  $O(n^2)$  for  $n$  labeled data points. Therefore, the large amount of calculation and enormous storage space caused by similarity graphs make these methods unscaleable to large-scale data sets. Second, these methods [25], [26], [27], [28], [29] utilize the pairwise similarities to guide the coding process; in other words, they preserve the point-wise similarities between data points. However, data belonging to the same category usually has shared attributes and discriminative attributes to distinguish them from data in other categories. When transforming labels into pairwise similarities, the data's category information is lost. Consequently, these methods neglect the data's intra-class similarities and inter-class dissimilarities; however, such information is important for retrieval.

To address these problems, a novel supervised multimodal hashing method called label consistent matrix factorization hashing (LCMFH) is proposed in this paper to directly use semantic labels to guide the hash learning procedure. The main concept of LCMFH is that heterogeneous data from the same semantic category shares attributes and can be represented by the same representations in the latent semantic spaces. In this way, hash codes quantified by the obtained representations are consistent with the semantic labels of the original data and, therefore, can have more discriminating power for cross-modal retrieval tasks. Fig. 1 illustrates the flowchart of the proposed LCMFH. It first jointly decomposes heterogeneous data matrices into latent semantic spaces and lets the unified representations be the linear combinations of semantic features with labels as coefficients. Next, hash codes are produced by quantizing the unified representations. In addition, this method learns linear projection matrices for out-of-sample extensions. Therefore, in the searching phase, linear projections can be directly utilized to generate binary hash codes. The main contributions of the proposed LCMFH are summarized as follows:

- A new matrix factorization method encouraging label consistency in latent semantic space is proposed for learning discriminative hash codes. It explicitly utilizes heterogeneous data along with their semantic labels to learn latent semantic spaces and guarantees that the corresponding unified representations are consistent with the semantic labels. Therefore, hash codes quantified by the unified representations better preserve the semantic similarities.
- By directly utilizing label information instead of the pairwise similarity matrix, the proposed method preserves the discriminative information for hash codes and thus gains more effectiveness as well as efficiency.
- Thorough experimental results demonstrate that the proposed method outperforms state-of-the-art supervised multimodal hashing methods in terms of both retrieval accuracy and scalability.

The rest of this paper is organized as follows: Section 2 introduces the proposed label consistent matrix factorization hashing model and its theoretical analysis. Section 3 presents the experimental results and comparisons using four real-world data sets. Finally, the conclusions are presented in Section 4.

## 2 LABEL CONSISTENT MATRIX FACTORIZATION HASHING

In this section, we present the details of the proposed LCMFH. First, the label consistent matrix factorization method is developed to uncover latent semantic spaces and unified representations. Second, hash functions are learned for the out-of-sample extensions. Finally, the overall objective function of LCMFH and the theoretical analysis are given.

To simplify the presentation, we first focus on hash code learning for bimodal data. Without loss of generality, it can be easily extended to cases with more modalities.

### 2.1 Notations and Problem Formulation

Suppose that  $\mathcal{O} = \{o_i\}_{i=1}^n$ ,  $o_i = (\mathbf{x}_i, \mathbf{y}_i)$  is a set of bimodal training entities, where  $\mathbf{x}_i \in \mathbb{R}^{d_x}$  and  $\mathbf{y}_i \in \mathbb{R}^{d_y}$  are two feature vectors from two modalities (e.g., image and text).  $d_x$  and  $d_y$  are the dimensionalities of the feature spaces in each modality, respectively, usually,  $d_x \neq d_y$ . These feature vectors form the columns of data matrices  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d_x \times n}$  and  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{d_y \times n}$ , respectively. Without loss of generality, we assume that data points are zero-centered, i.e.,  $\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$  and  $\sum_{i=1}^n \mathbf{y}_i = \mathbf{0}$ .

In addition to feature vectors, the semantic labels  $\mathbf{L} = [\mathbf{l}_1, \mathbf{l}_2, \dots, \mathbf{l}_n] \in \mathbb{R}^{c \times n}$  for training entities  $\mathcal{O}$  are available, where  $c$  is the total number of categories and  $\mathbf{l}_j = [l_{1j}, l_{2j}, \dots, l_{cj}]^T \in \mathbb{R}^c$  is the label vector for the  $j$ th training entity.  $l_{ij} = 1$  implies that the  $j$ th training entity belongs to the  $i$ th semantic category, otherwise  $l_{ij} = 0$ . Each label vector  $\mathbf{l}_j$  is normalized by  $\mathbf{l}_j = \mathbf{l}_j / \|\mathbf{l}_j\|_2$ .

The purpose of LCMFH is to learn unified binary codes  $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n] \in \mathbb{R}^{k \times n}$  for training entities  $\mathcal{O}$ , such that  $\mathbf{h}_i$  and  $\mathbf{h}_j$  preserve the semantic similarity given by labels between  $o_i$  and  $o_j$  with high probability, where  $\mathbf{h}_i \in \{0, 1\}^k$  and  $k$  is the length of the hash code. For out-of-sample instance, LCMFH learns the hash functions  $f(\mathbf{x})$  and  $f(\mathbf{y})$  for

each modality. In this paper, we utilize commonly used linear hash functions for their fast calculation abilities on large-scale data sets. Then, the hash functions are defined as follows

$$f(\mathbf{x}) = \text{sgn}(\mathbf{P}^x \mathbf{x}), \quad (1)$$

$$f(\mathbf{y}) = \text{sgn}(\mathbf{P}^y \mathbf{y}), \quad (2)$$

where  $\text{sgn}(\cdot)$  denotes the element-wise sign function and  $\mathbf{P}^x \in \mathbb{R}^{k \times d_x}$  and  $\mathbf{P}^y \in \mathbb{R}^{k \times d_y}$  are the linear projection matrices.

### 2.2 Label Consistent Matrix Factorization

It is well-known that multimedia data related to the same topic usually exists in different modalities, for instance, a news article often consists of text descriptions and images. Because relevant data from different modalities has semantic correlations, transforming original multimodal data into latent semantic spaces can maximize their cross-correlations and therefore improve cross-modal retrieval accuracy as shown by many existing studies [21], [22], [24], [30]. Matrix factorization, which learns a latent low dimensional space to adequately reconstruct the original data, is one of the most useful tools for learning latent information hidden in original data [31], [32]. Many unsupervised multimodal hashing methods use matrix factorization to transform data from the original feature space into latent semantic space. Collective matrix factorization hashing (CMFH) [22], latent semantic sparse hashing (LSSH) [21], and sparse multi-modal hashing [23] seek latent low dimensional spaces by matrix factorization to adequately reconstruct multimodal data and quantify the reconstruction coefficients to obtain the binary codes. Semantic topic multimodal hashing (STMH) [24] uses a robust version of matrix factorization to discover semantic concepts hidden in image. The good performance of these methods demonstrates the effectiveness of matrix factorization in multimodal hash learning applications.

However, the traditional matrix factorization methods are essentially unsupervised and cannot utilize label information. Therefore, it is inapplicable for the supervised multimodal hashing problem. To this end, we propose a new matrix factorization method called label consistent matrix factorization for the multimodal hash code learning task. It jointly utilizes heterogeneous data along with their semantic labels to learn basis vectors and representations. This method guarantees that data points sharing the same label will have the same latent semantic representations in the new reconstruction space. In this way, the learned semantic representations can have more discriminative power.

Given feature matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , using the corresponding semantic label matrix  $\mathbf{L}$ , the proposed label consistent matrix factorization jointly finds the basis vector matrices  $\mathbf{U}^x = [\mathbf{u}_1^x, \mathbf{u}_2^x, \dots, \mathbf{u}_k^x] \in \mathbb{R}^{d_x \times k}$  and  $\mathbf{U}^y = [\mathbf{u}_1^y, \mathbf{u}_2^y, \dots, \mathbf{u}_k^y] \in \mathbb{R}^{d_y \times k}$  associated with the unified representation matrix  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n] \in \mathbb{R}^{k \times n}$  to accurately approximate the original matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , e.g.,  $\mathbf{X} \approx \mathbf{U}^x \mathbf{V}$  and  $\mathbf{Y} \approx \mathbf{U}^y \mathbf{V}$ , where  $k$  is the length of the hash code. By matrix factorization, each feature vector is approximated by a linear combination of basis vectors, weighted by the corresponding representation. In fact, basis vector matrices  $\mathbf{U}^x$  and  $\mathbf{U}^y$  capture the higher-level features of the original data and can be

regarded as the basis vectors to form the latent semantic spaces hidden in multimodal data, and  $\mathbf{V}$  denotes the unified representations in latent semantic spaces.

To incorporate label information, we impose label constraints on  $\mathbf{V}$  by introducing an auxiliary matrix  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_c] \in \mathbb{R}^{k \times c}$  as  $\mathbf{V} = \mathbf{ZL}$ , where  $c$  is the number of categories. Each column  $\mathbf{z}_i$  in  $\mathbf{Z}$  is a coefficient to form the semantic information of the  $i$ th category by basis vectors. Therefore,  $\mathbf{z}_i$  is the unified semantic feature of the  $i$ th category hidden in the latent semantic spaces  $\mathbf{U}^x$  and  $\mathbf{U}^y$ . Each data value should contain the semantic feature of the categories it belongs to, i.e.,  $\mathbf{v}_j = \mathbf{Zl}_j = \sum_{i=1}^c l_{ij} \mathbf{z}_i$ . Therefore, the unified representations  $\mathbf{V}$  are the linear combinations of semantic features. Using the Frobenius norm as the cost function, the proposed label consistent matrix factorization can be formulated as

$$\begin{aligned} & \min_{\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}} \lambda \|\mathbf{X} - \mathbf{U}^x \mathbf{V}\|_F^2 + (1 - \lambda) \|\mathbf{Y} - \mathbf{U}^y \mathbf{V}\|_F^2 \\ & \text{s.t. } \mathbf{V} = \mathbf{ZL}, \end{aligned} \quad (3)$$

where  $\lambda \in (0, 1)$  is a parameter to balance the two parts.

Because the unified representations  $\mathbf{V}$  in (3) are closely related to the semantic labels, it is simple to confirm that when two data points share the same label (regardless of whether they belong to the same modality or different modalities), i.e.,  $\mathbf{l}_i = \mathbf{l}_j$ , and they will have the same representations in the new reconstruction spaces as  $\mathbf{v}_i = \mathbf{v}_j = \mathbf{Zl}_i = \mathbf{Zl}_j$ . Therefore, the intra-modal semantic similarity and inter-modal semantic similarity are both preserved by representations  $\mathbf{V}$ .

After obtaining unified representations  $\mathbf{V}$ , the hash codes can be obtained by quantifying  $\mathbf{V}$  as

$$\mathbf{H} = \text{sgn}(\mathbf{V}). \quad (4)$$

Therefore, the obtained hash codes also preserve the intra-modal semantic similarity and inter-modal semantic similarity.

Note that although CMFH, LSSH, STMH, and the proposed LCMFH all use matrix factorization in their learning processes, their main concepts are different. First, LCMFH is a supervised method whereas the others are unsupervised methods. LCMFH jointly utilizes heterogeneous data along with their semantic labels to learn hash codes. Its main concept is to guarantee data points with the same labels have the same hash codes. That is, LCMFH intends to preserve the semantic similarities given by the labels. In contrast, CMFH, LSSH and STMH learn hash codes from data distributions to preserve the Euclidean similarities of the training data. Second, LCMFH utilizes matrix factorization for all the modalities and combines with a constraint to guarantee data points sharing the same label have the same representations in the new reconstruction space. CMFH utilizes collective matrix factorization to learn unified hash codes for all the modalities. STMH learns binary codes for one modality and discovers representations for other modalities by matrix factorization to improve robustness against noisy and unreliable data. LSSH uses matrix factorization to learn hash codes for images. Therefore, the proposed LCMFH method is different from other matrix factorization based multimodal hashing methods.

### 2.3 Hash Function Learning

The unified representations  $\mathbf{V}$  for training data can be obtained directly based on (3), but it cannot be generalized to query directly. For out-of-sample instances, we learn two linear hash functions that map the feature vectors of images and texts to unified representations by

$$\min_{\mathbf{P}^x, \mathbf{P}^y, \mathbf{V}} \|\mathbf{V} - \mathbf{P}^x \mathbf{X}\|_F^2 + \|\mathbf{V} - \mathbf{P}^y \mathbf{Y}\|_F^2. \quad (5)$$

When a new query arrives, its hash code can be obtained by the corresponding projection matrix  $\mathbf{P}^x$  or  $\mathbf{P}^y$ .

### 2.4 Overall Objective Function

The overall objective function of LCMFH combines the label consistent matrix factorization term given in (3), the hash function learning term given in (5) and the regularization term is defined as

$$\begin{aligned} & \min_{\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}, \mathbf{P}^x, \mathbf{P}^y} \mathcal{F}(\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}, \mathbf{P}^x, \mathbf{P}^y) \\ & \text{s.t. } \mathbf{V} = \mathbf{ZL}, \end{aligned} \quad (6)$$

where

$$\begin{aligned} \mathcal{F} = & \lambda \|\mathbf{X} - \mathbf{U}^x \mathbf{V}\|_F^2 + (1 - \lambda) \|\mathbf{Y} - \mathbf{U}^y \mathbf{V}\|_F^2 \\ & + \mu \left( \|\mathbf{V} - \mathbf{P}^x \mathbf{X}\|_F^2 + \|\mathbf{V} - \mathbf{P}^y \mathbf{Y}\|_F^2 \right) \\ & + \gamma R(\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}, \mathbf{P}^x, \mathbf{P}^y), \end{aligned} \quad (7)$$

where  $\lambda$ ,  $\mu$ , and  $\gamma$  are nonnegative tradeoff parameters controlling the relative contributions of the corresponding terms, and the regularization term is defined as  $R(\cdot) = \|\cdot\|_F^2$  to avoid overfitting.

In objective function (6), we jointly learn  $\mathbf{P}^x$ ,  $\mathbf{P}^y$  and other parameters to narrow the gap between the hash codes of the training data and the query data. In the training stage, the training data's hash codes are obtained by quantizing  $\mathbf{V}$ . Additionally in the query stage, the query data's hash codes are obtained by quantizing the projection representations  $\mathbf{P}^x \mathbf{X}$  or  $\mathbf{P}^y \mathbf{Y}$ . There is a gap between the hash codes of the training data and the query data. In a joint learning manner, all the variables interact with each other. By minimizing the main objective function (6), the reconstruction error  $\lambda \|\mathbf{X} - \mathbf{U}^x \mathbf{V}\|_F^2 + (1 - \lambda) \|\mathbf{Y} - \mathbf{U}^y \mathbf{V}\|_F^2$  and the projection error  $\mu (\|\mathbf{V} - \mathbf{P}^x \mathbf{X}\|_F^2 + \|\mathbf{V} - \mathbf{P}^y \mathbf{Y}\|_F^2)$  are minimized simultaneously. Therefore, the gap can be narrowed and the retrieval performance will be improved. If we learn  $\mathbf{P}^x$  and  $\mathbf{P}^y$  separately, the reconstruction error and the projection error are minimized separately and, may not be optimal globally. Therefore, we jointly learn all the parameters in LCMFH.

### 2.5 Optimization Method

The optimization problem of (7) is non-convex with the matrix variables  $\mathbf{U}^x$ ,  $\mathbf{U}^y$ ,  $\mathbf{P}^x$ ,  $\mathbf{P}^y$ , and  $\mathbf{V}$ . Fortunately, it is convex with respect to any one of the five matrix variables if the others are fixed. In the following, we introduce an iterative algorithm that can reach local minima.

We first rewrite the objective function  $\mathcal{F}$  as

$$\begin{aligned}\mathcal{F} = & \lambda \|\mathbf{X} - \mathbf{U}^x \mathbf{ZL}\|_F^2 + (1 - \lambda) \|\mathbf{Y} - \mathbf{U}^y \mathbf{ZL}\|_F^2 \\ & + \mu (\|\mathbf{ZL} - \mathbf{P}^x \mathbf{X}\|_F^2 + \|\mathbf{ZL} - \mathbf{P}^y \mathbf{Y}\|_F^2) \\ & + \gamma R(\mathbf{U}^x, \mathbf{U}^y, \mathbf{ZL}, \mathbf{P}^x, \mathbf{P}^y).\end{aligned}\quad (8)$$

Next we solve the following steps iteratively to obtain the updating rules.

- 1) Fixing  $\mathbf{U}^y, \mathbf{V}, \mathbf{P}^x$ , and  $\mathbf{P}^y$ , and letting the derivative of  $\mathcal{F}$  with respect to  $\mathbf{U}^x$  equal zero, we obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}^x} = 2\lambda \mathbf{U}^x \mathbf{ZL} \mathbf{L}^T \mathbf{Z}^T + 2\gamma \mathbf{U}^x - 2\lambda \mathbf{XL}^T \mathbf{Z}^T = 0. \quad (9)$$

Then, we can obtain the closed form solution of  $\mathbf{U}^x$

$$\mathbf{U}^x = \mathbf{XL}^T \mathbf{Z}^T \left( \mathbf{ZL} \mathbf{L}^T \mathbf{Z}^T + \frac{\gamma}{\lambda} \mathbf{I} \right)^{-1}, \quad (10)$$

where  $\mathbf{I}$  is the identity matrix.

- 2) Fixing  $\mathbf{U}^x, \mathbf{V}, \mathbf{P}^x$ , and  $\mathbf{P}^y$ , and letting the derivative of  $\mathcal{F}$  with respect to  $\mathbf{U}^y$  equal zero, we obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{U}^y} = 2(1 - \lambda) (\mathbf{U}^y \mathbf{ZL} \mathbf{L}^T \mathbf{Z}^T - \mathbf{YL}^T \mathbf{Z}^T) + 2\gamma \mathbf{U}^y = 0. \quad (11)$$

Then, we can obtain the closed form solution of  $\mathbf{U}^y$

$$\mathbf{U}^y = \mathbf{YL}^T \mathbf{Z}^T \left( \mathbf{ZL} \mathbf{L}^T \mathbf{Z}^T + \frac{\gamma}{1 - \lambda} \mathbf{I} \right)^{-1}. \quad (12)$$

- 3) Fixing  $\mathbf{U}^x, \mathbf{U}^y, \mathbf{P}^x$ , and  $\mathbf{P}^y$ , and letting the derivative of  $\mathcal{F}$  with respect to  $\mathbf{Z}$  equal zero, we obtain

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{Z}} = & 2\lambda \mathbf{U}^{xT} \mathbf{U}^x \mathbf{ZL}^T + 2(1 - \lambda) \mathbf{U}^{yT} \mathbf{U}^y \mathbf{ZL}^T \\ & + 2(2\mu + \gamma) \mathbf{ZL}^T - 2\mu \mathbf{P}^x \mathbf{XL}^T - 2\mu \mathbf{P}^y \mathbf{YL}^T \\ & - 2\lambda \mathbf{U}^{xT} \mathbf{XL}^T - 2(1 - \lambda) \mathbf{U}^{yT} \mathbf{YL}^T = 0.\end{aligned}\quad (13)$$

Then, we can obtain the closed form solution of  $\mathbf{Z}$

$$\begin{aligned}\mathbf{Z} = & [\lambda \mathbf{U}^{xT} \mathbf{U}^x + (1 - \lambda) \mathbf{U}^{yT} \mathbf{U}^y + (2\mu + \gamma) \mathbf{I}]^{-1} \\ & \times [\mu (\mathbf{P}^x \mathbf{X} + \mathbf{P}^y \mathbf{Y}) \mathbf{L}^T + \lambda \mathbf{U}^{xT} \mathbf{XL}^T + (1 - \lambda) \mathbf{U}^{yT} \mathbf{YL}^T] \\ & \times (\mathbf{L}^T)^{-1}.\end{aligned}\quad (14)$$

- 4) Fixing  $\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}$ , and  $\mathbf{P}^y$ , and letting the derivative of  $\mathcal{F}$  with respect to  $\mathbf{P}^x$  equal zero, we obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{P}^x} = 2\mu \mathbf{P}^x \mathbf{XX}^T + 2\gamma \mathbf{P}^x - 2\mu \mathbf{ZLX}^T = 0. \quad (15)$$

Then, we can obtain the closed form solution of  $\mathbf{P}^x$

$$\mathbf{P}^x = \mathbf{ZLX}^T \left( \mathbf{XX}^T + \frac{\gamma}{\mu} \mathbf{I} \right)^{-1}. \quad (16)$$

- 5) Fixing  $\mathbf{U}^x, \mathbf{U}^y, \mathbf{V}$ , and  $\mathbf{P}^x$ , and letting the derivative of  $\mathcal{F}$  with respect to  $\mathbf{P}^y$  equal zero, we obtain

$$\frac{\partial \mathcal{F}}{\partial \mathbf{P}^y} = 2\mu \mathbf{P}^y \mathbf{YY}^T + 2\gamma \mathbf{P}^y - 2\mu \mathbf{ZLY}^T = 0. \quad (17)$$

Then, we can obtain the closed form solution of  $\mathbf{P}^y$

$$\mathbf{P}^y = \mathbf{ZLY}^T \left( \mathbf{YY}^T + \frac{\gamma}{\mu} \mathbf{I} \right)^{-1}. \quad (18)$$

Therefore, the objective function  $\mathcal{F}$  can be solved by updating the above steps iteratively until convergence or until a preset maximum number of iterations has been reached. For the first round, all the matrix variables are initialized with the matrices of normally distributed random numbers. The algorithm is summarized in Algorithm 1.

---

**Algorithm 1.** Label Consistent Matrix Factorization Hashing

---

**Training stage**

**Input:** feature matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , semantic labels  $\mathbf{L}$ , code length  $k$ , parameters  $\lambda, \mu$  and  $\gamma$ .

**Output:** intergrated hash codes  $\mathbf{H}$ , linear projection matrices  $\mathbf{P}^x$  and  $\mathbf{P}^y$ .

**Procedure:**

1. Centering  $\mathbf{X}$  and  $\mathbf{Y}$  by means.
2. Normalize semantic labels  $\mathbf{L}$ .
3. Initialize  $\mathbf{Z}$  by random matrix.

**Repeat**

- 4.1 Update  $\mathbf{U}^x$  by Eq. (10).
- 4.2 Update  $\mathbf{U}^y$  by Eq. (12).
- 4.3 Update  $\mathbf{P}^x$  by Eq. (16).
- 4.4 Update  $\mathbf{P}^y$  by Eq. (18).
- 4.5 Update  $\mathbf{Z}$  by Eq. (14).

**Until** Stopping criteria is met.

5. Calculate hash codes by  $\mathbf{H} = \text{sgn}(\mathbf{ZL})$ .

**Testing stage**

**Input:** feature vector  $\mathbf{x}^t$  or  $\mathbf{y}^t$ , linear projection matrices  $\mathbf{P}^x$  and  $\mathbf{P}^y$ .

**Output:** hash code  $\mathbf{h}^t$ .

**Procedure:**

1. Centering  $\mathbf{x}^t$  or  $\mathbf{y}^t$  by means.
  2. For  $\mathbf{x}^t$  : calculate hash code by  $\mathbf{h}^t = \text{sgn}(\mathbf{P}^x \mathbf{x}^t)$ .  
For  $\mathbf{y}^t$  : calculate hash code by  $\mathbf{h}^t = \text{sgn}(\mathbf{P}^y \mathbf{y}^t)$ .
- 

We present the following theorem regarding the above iterative updating rules:

**Theorem 1.** Algorithm 1 decreases the objective value of (8) in each iteration.

Please see Appendix A, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2861000>, for a detailed proof of Theorem 1. It guarantees the convergence of the iterations in Algorithm 1; therefore, the final solution will be a local optimum.

## 2.6 Complexity Analysis

In this section, we detail the time complexity of LCMFH and show that it is applicable to large-scale multimodal data sets. The objective function is minimized by iteratively updating the variables. Therefore, the computational complexity is very important for the algorithm efficiency. Another factor is the convergence rate. We will discuss the computational complexity of the updating algorithms in this section and experimentally demonstrate the convergence rate in the experiments section.

In the training phase, the time complexity of each step is  $O(k^3 + c^3 + d^3 + (d + c)k^2 + (dk + kc + k^2 + c^2 + d^2)n)$ , where  $d = \max\{d_x, d_y\}$ . Because  $k, d$ , and  $c \ll n$ , the training complexity is  $O(m^2n)$ , where  $m = \max\{d, c, k\}$ , which is linear to the training set size. Given the iteration number  $T$ , the overall training complexity for LCMFH is  $O(m^2nT)$ . Usually,  $T$  is very small and this will be verified in the following experiments section. In fact, training LCMFH is much faster compared with most existing multimodal hashing methods.

In the search phase, the complexity is  $O(dk)$ . The query time complexity of LCMFH is also very low.

In summary, the training complexity of LCMFH is linear to  $n$  and the query complexity of LCMFH is constant. Hence, the proposed LCMFH is highly scalable for large-scale data sets.

## 2.7 Multiple Modalities Extension

The extension of LCMFH in (6) from bimodal to multiple modalities is quite simple and direct:

$$\begin{aligned} \min_{\mathbf{U}^t, \mathbf{P}^t, \mathbf{V}} \mathcal{F} &= \sum_t \lambda_t \|\mathbf{X}^t - \mathbf{U}^t \mathbf{V}\|_F^2 + \mu \sum_t \|\mathbf{V} - \mathbf{P}^t \mathbf{X}^t\|_F^2 \\ &\quad + \gamma \left( \sum_t R(\mathbf{U}^t, \mathbf{P}^t) + R(\mathbf{V}) \right) \\ \text{s.t. } \mathbf{V} &= \mathbf{ZL}, \sum_t \lambda_t = 1, \end{aligned} \quad (19)$$

where  $\mathbf{X}^t$  is the feature matrix of each modality. It is straightforward to adapt Algorithm 1 to solve the new optimization problem.

## 2.8 Semi-Supervised Extension

The proposed label consistent matrix factorization hashing is a supervised method. It uses all the training data and their labels to learn the hash functions. In some cases, the training data is only partially labeled. To address, we extend the proposed method to a semi-supervised setting. For semi-supervised label consistent matrix factorization hashing (SLCMFH), we first use the labeled data to train the hash functions, and then generate hash codes for both the training set and query set using the learned hash functions.

## 3 EXPERIMENTS

To evaluate the performance of the proposed LCMFH, we conduct cross-modal retrieval experiments with several state-of-the-art methods on four benchmark data sets, i.e., Wiki,<sup>1</sup> MIRFlickr,<sup>2</sup> NUS-WIDE,<sup>3</sup> and MSCOCO.<sup>4</sup> Thus far, these are the largest publicly available multimodal data sets that are fully paired and labeled. The statistics of these data sets are shown in Table 1. Despite multimodal hashing methods are mainly designed for cross-modal retrieval, they can also be readily used to perform unimodal retrieval by the learned hash codes. Therefore, we report unimodal retrieval results of these methods as well.

TABLE 1  
Statistics of Benchmark Data Sets

Data set	Dimensionality		Class	Size
	Image	Text		
Wiki	128	10	10	2,866
MIRFlickr	150	500	20	20,015
NUS-WIDE	128	1,000	10	186,577
MSCOCO	512	512	80	122,218

### 3.1 Data Sets

Wiki contains 2,866 image-text pairs collected from Wikipedia's articles. Specifically, each pair is labeled with one of 10 semantic classes. Following the approach in [33], each image is first represented as a 4,096-dimensional feature extracted by the Caffe implementation of AlexNet [34] and then is represented as a 128-dimensional feature by performing PCA on the extracted features. Each text is represented by the probability distribution over 10 topics learned by a latent Dirichlet allocation (LDA) model. The data set is released with a training set with 2,173 pairs and a query set with 693 pairs. Because the data set is fully annotated, the semantic neighbors based on the associated labels, are treated as the ground truth for a query.

MIRFlickr contains 25,000 images downloaded from the Flickr website. Each image is associated with several user-assigned tags and is manually annotated with some of the 24 provided unique labels. Following the pretreatment in [28], we select the textual tags that appear at least 20 times and remove image-tag pairs without textual tags or manually annotated labels. Then, 20,015 image-tag pairs can be obtained. For each pair, an image is represented as a 150-dimensional edge histogram feature vector and text is represented as a 500-dimensional feature vector derived from the PCA on the index vector of the textual tags. We randomly select 5 percent of the image-tag pairs as the query set and the left pairs are treated as the training set. True semantic neighbors for a query are defined as those sharing at least one label with the query.

NUS-WIDE is a real-world web image data set containing 269,684 images downloaded from Flickr. Tagging ground-truth for 81 semantic concepts is provided for evaluation. Each image is first represented as a 4,096-dimensional feature extracted by the Caffe implementation of VGG Net [35] and then is represented as a 128-dimensional feature by performing PCA on the extracted features, and each text is represented by an index vector of the most frequent 1,000 tags. Following the settings in [22], [29], we select the ten largest concepts and the corresponding 186,577 images for experiment. Of these, 1 percent of the images with their tags are randomly chosen to serve as the query set and the remaining image-tag pairs serve as the training set. True semantic neighbors for a query are defined as those sharing at least one label with the query.

MSCOCO is a new image recognition, segmentation, and captioning data set containing more than 300,000 images. The data set has several features: 80 object categories, multiple labels per image, 5 captions per image, etc. In our experiments, images are first represented as 4,096-dimensional features extracted by the Caffe implementation

1. <http://www.svcl.ucsd.edu/projects/crossmodal/>  
 2. <http://press.liacs.nl/mirflickr/mirdownload.html>  
 3. <http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm>  
 4. <http://mscoco.org/home/>

of VGG Net and then are represented as 512-dimensional features by performing PCA on the extracted features. The corresponding captions are treated as texts and are represented by 20,629-dimensional bag-of-words feature vectors by natural language toolkit (NLTK). Next, the text features are represented as 512-dimensional features by PCA. We remove images without captions or labels. Then, 122,218 image-text pairs can be obtained. We randomly take 2,000 image-text pairs as the query set, and the left pairs are treated as the training set. True semantic neighbors for a query are defined as those sharing at least one label with the query.

### 3.2 Evaluation Metrics

One common performance measure is the mean of average precision (mAP). Given a query and a list of  $R$  retrieved results, its average precision (AP) is defined as

$$AP = \frac{1}{N} \sum_{r=1}^R P(r)\delta(r), \quad (20)$$

where  $N$  is the number of relevant instances in the retrieved set,  $P(r)$  denotes the precision of the top  $r$  retrieved instances, and  $\delta(r) = 1$  if the  $r$ th retrieved instances is a true neighbor of the query; otherwise,  $\delta(r) = 0$ . Then, the AP of all queries are averaged to obtain the mAP. Clearly, the larger the mAP is, the better the performance is.  $R$  is set to 100 in the following experiments.

We also report two other types of measures: including Precision-Recall curve, which can be obtained by varying the Hamming radius of the retrieved points and evaluating the precision and recall accordingly; and topN-precision, which reflects the change of precision with respect to the number of retrieved instances. Generally, a higher topN-Precision curve or Precision-Recall curve indicates better performance. A detailed description of the above evaluation metrics can be referred to in [36].

### 3.3 Baseline Methods

We compare the LCMFH model against a range of state-of-the-art multimodal hashing methods, from the pioneering work CMSSH [25] to the most recent work SEPH [28], using the evaluation metrics introduced above.

- The CMSSH method [25] embeds multimodal data into a common Hamming space by eigenvalue decomposition and boosting.
- The CVH method [26] extends traditional spectral hashing to the multimodal case by mapping similar objects across different modalities to similar hash codes.
- The MLBE method [27] employs a generative probabilistic model to learn binary codes according to the estimation of the maximum a posteriori.
- The SEPH [28] transforms a pairwise similarity matrix into a probability distribution and approximates it with to-be-learnt hash codes by minimizing the KLD.
- The SCM method [29] aims to reconstruct the similarity matrix by the learned hash code. In the following experiments, we use the sequential learning method

of SCM for its superior performance compared with the orthogonal projection learning method of SCM.

- The CMFH method [22] learns unified hash codes by collective matrix factorization with latent factor model from different modalities of one instance.
- The STMH method [24] models text as multiple semantic topics and images as latent semantic concepts, and generates hash codes by probing whether topics or concepts are available in the data.

Among these above seven methods, CMFH and STMH are unsupervised methods whereas the others are supervised methods. The reason for comparing the proposed LCMFH with CMFH and STMH is because they all use matrix factorization strategies to learn hash codes.

### 3.4 Implementation Details

CMSSH, MLBE and SEPH have high computational costs. It is quite difficult to learn hash functions on a large data set using all the training data. Thus, we randomly select 10,000 instances from the training set for these methods to train hash functions and then apply the trained hash functions to the other instances in the training set to generate hash codes, similar to [22]. MLBE requires much computational resource when the code length is large even on small data sets. For this reason, we do not report its performance with a 128-bit code length.

In the parameter sensitivity analysis section, we present an empirical analysis of the parameter sensitivity, which verifies that LCMFH can achieve a stable performance under a wide range of parameter values. When comparing with the baseline methods, we use the following parameter settings,  $\lambda = 0.5$ ,  $\mu = 10$ , and  $\gamma = 0.001$ . For all the baseline algorithms except CVH, the codes have been kindly provided by the authors. Since the original implementation of CVH is not publicly available, we carefully implement it by ourselves. We tune the parameters of all the baseline algorithms according to the corresponding studies. The convergence threshold and maximum numbers of iterations are set to 0.01 and 20, respectively, for LCMFH, CMFH and STMH in the following experiments.

All the baseline methods and the proposed LCMFH are implemented using MATLAB. The experiments are conducted on a personal computer with an Intel Xeon(R) running at 2.0 GHz with 8 cores, 64 GB RAM and a 64-bit GNU/Linux operating system.

### 3.5 Results and Discussions

#### 3.5.1 Results on Wiki

The mAP values of the LCMFH and seven baseline methods on the Wiki data set are reported in Table 2. The topN-precision curves, and precision-recall curves are plotted in Figs. 2 and 3, respectively.

From Table 2, we have the following four observations. First, SEPH always yields the best mAP results compared with other methods. Second, the mAP values of the proposed LCMFH are slightly lower than those of the SEPH but are much better than those of the others. Third, the mAP values of the image query are quite low compared with those of the text query for all the methods. The last observation is that the performance of the unsupervised CMFH and STMH is even

**TABLE 2**  
mAP Results on Different Data Sets

Task	Method	Wiki				MIRflickr				NUS-WIDE				MSCOCO			
		16 bits	32 bits	64 bits	128 bits	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	CMFH	0.2559	0.3190	0.3381	0.3509	0.6434	0.6333	0.6208	0.6163	0.7782	0.8191	0.8394	0.8485	0.7274	0.8007	0.8371	0.8574
	STMH	0.3669	0.4033	0.4115	0.4266	0.6033	0.6305	0.6271	0.5817	0.7151	0.7526	0.7832	0.8050	0.4465	0.5459	0.5629	0.6409
	CMSSH	0.2787	0.2826	0.2798	0.2845	0.5275	0.5532	0.5577	0.6162	0.7210	0.7113	0.7512	0.8012	0.5531	0.6592	0.6810	0.7329
	CVH	0.3130	0.2937	0.2722	0.2622	0.5761	0.5787	0.5843	0.6374	0.8100	0.8231	0.8195	0.7983	0.6470	0.7122	0.7269	0.7275
	MLBE	0.1484	0.1508	0.1510	-	0.5702	0.5089	0.5335	-	0.5668	0.3720	0.5025	-	0.4093	0.3180	0.3968	-
	SCM	0.3362	0.3377	0.3500	0.3566	0.6576	0.6627	0.6565	0.6703	0.7985	0.8280	0.8341	0.8377	0.7306	0.8095	0.8448	0.8621
	SEPH	<b>0.4420</b>	<b>0.4670</b>	<b>0.4701</b>	<b>0.4831</b>	0.6467	0.6471	0.6525	0.6536	0.8428	0.8539	0.8591	0.8691	0.7885	0.8415	0.8715	0.8918
Text query	LCMFH	0.4101	0.4136	0.4261	0.4387	<b>0.7870</b>	<b>0.8155</b>	<b>0.8254</b>	<b>0.8376</b>	<b>0.9216</b>	<b>0.9386</b>	<b>0.9480</b>	<b>0.9541</b>	<b>0.8184</b>	<b>0.8762</b>	<b>0.9147</b>	<b>0.9343</b>
	CMFH	0.3657	0.4127	0.4376	0.4627	0.7090	0.7229	0.7477	0.7697	0.7289	0.7554	0.7743	0.7748	0.6001	0.5515	0.5769	0.5786
	STMH	0.5948	0.6154	0.6242	0.6371	0.7058	0.7403	0.7536	0.7633	0.6550	0.6841	0.6986	0.7066	0.4500	0.4859	0.5442	0.5901
	CMSSH	0.5930	0.6077	0.6128	0.6183	0.7225	0.7461	0.7544	0.7563	0.6305	0.6646	0.6810	0.6928	0.5266	0.5205	0.5357	0.5597
	CVH	0.5941	0.5760	0.5648	0.5317	0.7926	0.8309	0.8403	0.8464	0.6782	0.6801	0.6862	0.6758	0.6396	0.6601	0.6674	0.6752
	MLBE	0.4128	0.5174	0.5868	-	0.5913	0.5824	0.6029	-	0.3993	0.4009	0.3924	-	0.4392	0.3496	0.4603	-
	SCM	0.6018	0.5979	0.6199	0.6289	0.7994	0.8277	0.8373	0.8497	0.7275	0.7504	0.7535	0.7594	0.6048	0.6562	0.6872	0.7062
Text	SEPH	<b>0.6783</b>	<b>0.6870</b>	<b>0.6896</b>	<b>0.6880</b>	0.8037	0.8340	0.8480	0.8568	0.7163	0.7203	0.7242	0.7263	0.6758	0.7261	0.7415	0.7606
	LCMFH	0.6369	0.6386	0.6568	0.6706	<b>0.9047</b>	<b>0.9209</b>	<b>0.9311</b>	<b>0.9338</b>	<b>0.8677</b>	<b>0.8783</b>	<b>0.8930</b>	<b>0.8999</b>	<b>0.7405</b>	<b>0.7339</b>	<b>0.8094</b>	<b>0.8165</b>
Image	CMFH	0.2559	0.3190	0.3381	0.3509	0.6434	0.6333	0.6208	0.6163	0.7782	0.8191	0.8394	0.8485	0.7274	0.8007	0.8371	0.8574
	STMH	0.3669	0.4033	0.4115	0.4266	0.6033	0.6305	0.6271	0.5817	0.7151	0.7526	0.7832	0.8050	0.4465	0.5459	0.5629	0.6409
	CMSSH	0.2943	0.2852	0.2460	0.2456	0.6523	0.6460	0.6756	0.6422	0.5652	0.5976	0.6021	0.5500	0.5815	0.6156	0.5974	0.5719
	CVH	0.1141	0.1169	0.1262	0.1297	0.5859	0.5843	0.5881	0.6447	0.3568	0.4195	0.3211	0.3530	0.6299	0.6100	0.5853	0.5179
	MLBE	0.1678	0.1458	0.1504	-	0.5135	0.5202	0.5817	-	0.4319	0.4404	0.3919	-	0.4773	0.3792	0.4657	-
	SCM	0.3840	0.3926	0.3982	0.4080	0.6884	0.6992	0.6981	0.7125	0.7847	0.8084	0.8093	0.8131	0.7105	0.7888	0.8263	0.8414
	SEPH	<b>0.4420</b>	<b>0.4670</b>	<b>0.4701</b>	<b>0.4831</b>	0.6467	0.6471	0.6525	0.6536	0.8428	0.8539	0.8591	0.8691	0.7885	0.8415	0.8715	0.8918
Text	LCMFH	0.4101	0.4136	0.4261	0.4387	<b>0.7870</b>	<b>0.8155</b>	<b>0.8254</b>	<b>0.8376</b>	<b>0.9216</b>	<b>0.9386</b>	<b>0.9480</b>	<b>0.9541</b>	<b>0.8184</b>	<b>0.8762</b>	<b>0.9147</b>	<b>0.9343</b>
Text	CMFH	0.3657	0.4127	0.4376	0.4627	0.7090	0.7229	0.7477	0.7697	0.7289	0.7554	0.7743	0.7748	0.6001	0.5515	0.5769	0.5786
	STMH	0.5948	0.6154	0.6242	0.6371	0.7058	0.7403	0.7536	0.7633	0.6550	0.6841	0.6986	0.7066	0.4500	0.4859	0.5442	0.5901
	CMSSH	0.4025	0.4072	0.3830	0.3493	0.5904	0.5670	0.5559	0.5726	0.5826	0.5040	0.5061	0.4685	0.5345	0.3398	0.4080	0.3666
	CVH	0.1195	0.1154	0.1130	0.1312	0.5793	0.5960	0.6172	0.6450	0.3438	0.3520	0.2756	0.2982	0.6145	0.5655	0.4589	0.3981
	MLBE	0.1416	0.1503	0.1456	-	0.5534	0.4719	0.5726	-	0.4760	0.4401	0.4079	-	0.5400	0.2870	0.4044	-
	SCM	0.4846	0.4852	0.4881	0.5078	0.6927	0.7038	0.7096	0.7225	0.7298	0.7518	0.7545	0.7583	0.5712	0.6042	0.6459	0.6953
	SEPH	<b>0.6783</b>	<b>0.6870</b>	<b>0.6896</b>	<b>0.6880</b>	0.8037	0.8340	0.8480	0.8568	0.7163	0.7203	0.7247	0.7263	0.6758	0.7261	0.7415	0.7606
Image	LCMFH	0.6369	0.6386	0.6568	0.6706	<b>0.9047</b>	<b>0.9209</b>	<b>0.9311</b>	<b>0.9338</b>	<b>0.8677</b>	<b>0.8783</b>	<b>0.8930</b>	<b>0.8999</b>	<b>0.7405</b>	<b>0.7339</b>	<b>0.8094</b>	<b>0.8165</b>

The best performance is shown in boldface. '-' denotes an untested value under that specific setting.

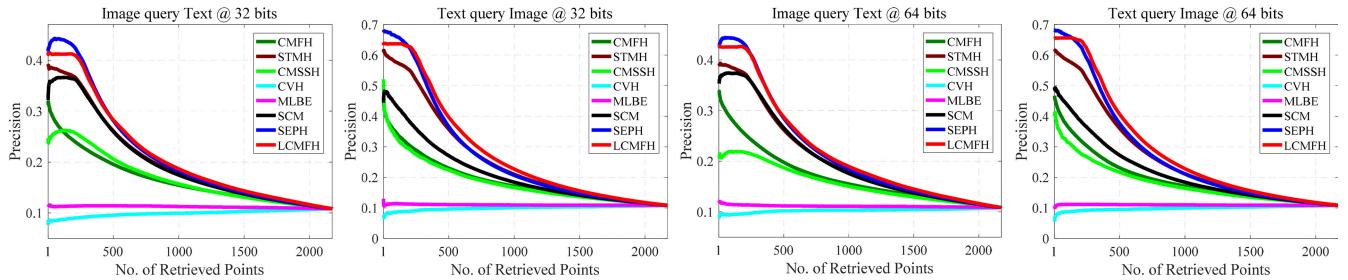


Fig. 2. topN-Precision curves on Wiki by varying code length.

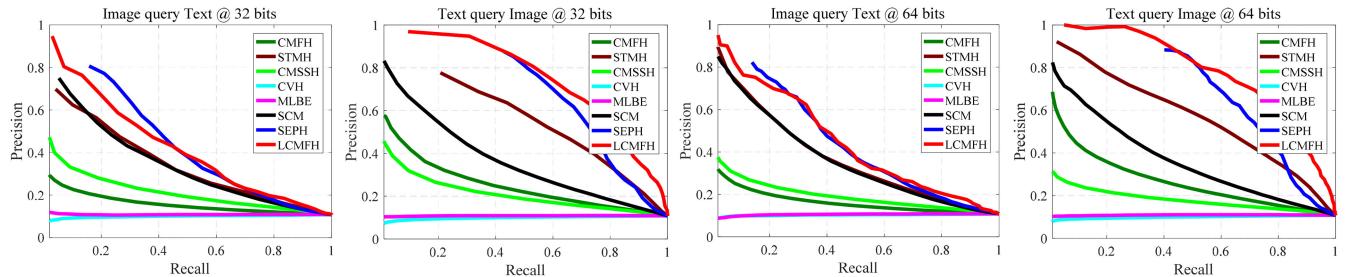


Fig. 3. Precision-Recall curves on Wiki by varying code length.

better than that of the supervised CMSSH, CVH and MLBE methods. There are two possible reasons for these observations. First, the semantic gap between the two modalities of Wiki is quite large [21]. The images are not quite related to

their texts. Therefore, it is sometimes difficult to find semantically similar texts or images by image query, resulting in a poor search accuracy for both image-query-image and image-query-text tasks. Second, some images are not closely

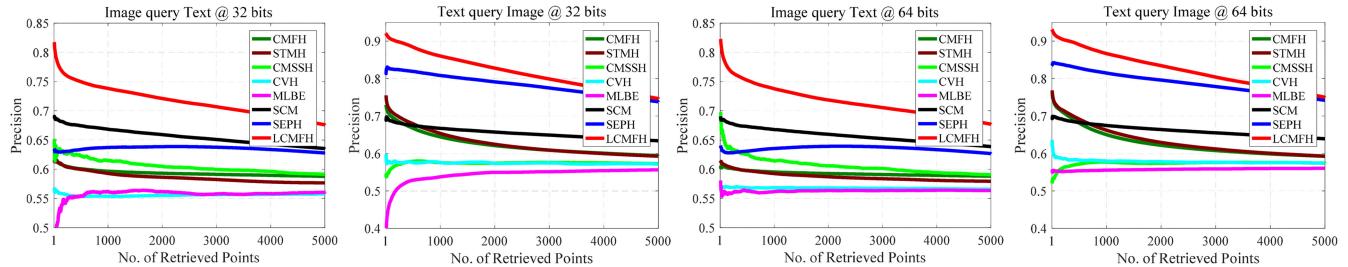


Fig. 4. topN-Precision curves on MIRFlickr by varying code length.

related to their assigned categories. In such cases, the extracted image features can not reflect the semantic properties appropriately, which leads to inconsistencies between the image features and the semantic label information. Because SEPH uses only semantic similarities without considering features to learn hash codes, it avoids the inconsistencies between image features and semantic information and, therefore, has better mAP values than other methods. The reason for the fourth observation may also be due to the inconsistencies between image features and semantic information. Because CMFH and STMH are unsupervised methods, they use only image features to learn the hash codes. In contrast, supervised CMSSH, CVH and MLBE are affected by inaccurate semantic information and thus achieve much lower mAP values.

Fig. 2 plots the topN-precision curves for all the methods. We can observe that LCMFH always achieves the second-best performance among all the baseline methods, which is consistent with the mAP evaluation. The precision-recall curves are plotted in Fig. 3, which shows that LCMFH achieves the best performance compared with the baseline methods on text-query-image similarity search task. The performance of LCMFH on image-query-text task is comparable to that of SEPH.

In general, the proposed LCMFH achieves satisfactory performance with different code lengths under different evaluation criteria on the Wiki data set. Although its performance is slightly lower than SEPH with regard to mAP values, LCMFH requires much less training time than SEPH, which will be verified in the running time section.

### 3.5.2 Results on MIRFlickr

The mAP values of LCMFH and seven baseline methods on the MIRFlickr data set are reported in Table 2. LCMFH achieves a substantial improvement of 10 percent over the second-best method (i.e., SCM or SEPH). In addition, all the methods achieve higher mAP values with the increase of the code length. This is reasonable because longer hash

codes can encode more semantic information. Moreover, LCMFH achieves a much better performance with the smallest code length (16 bits) than other methods do with the largest code length (128 bits). This indicates that LCMFH preserves more semantic information than the other methods do even with fewer bits.

The topN-precision curves, and precision-recall curves of all the methods on the MIRFlickr data set are plotted in Figs. 4 and 5, respectively. We can observe that LCMFH significantly outperforms all baseline methods on all the tasks with different code lengths, which verifies the effectiveness of LCMFH.

We clearly see from these results that all the baseline methods produce better results on the MIRFlickr data set in contrast to the poor performance on the Wiki data set particularly on image-query-image and image-query-text tasks. A possible reason for this phenomenon is that the semantic gap between the two modalities of MIRFlickr is smaller than that of Wiki.

To let readers have an intuitive feeling of retrieval results, we show some visualization results of cross-modal retrieval on the MIRFlickr data set in Figs. 6 and 7, respectively. The hash code length is set to 32 bits. For each text query, we show top 6 nearest images according to Hamming distances of their hash codes to the hash code of the textual query in Fig. 6. We can see that the retrieved images of LCMFH are more semantically relevant to the query texts than other methods. Similarly, for the image query, we show top 30 nearest tags in Fig. 7. As there are some irrelevant tags, we manually judge whether the returned tags are relevant to the query image and mark them in blue. We can observe that LCMFH usually gives more semantically relevant tags to the image query. This shows the superior performance of LCMFH than baselines.

### 3.5.3 Results on NUS-WIDE

The mAP values of LCMFH and seven baseline methods on the NUS-WIDE data set are reported in Table 2. The topN-

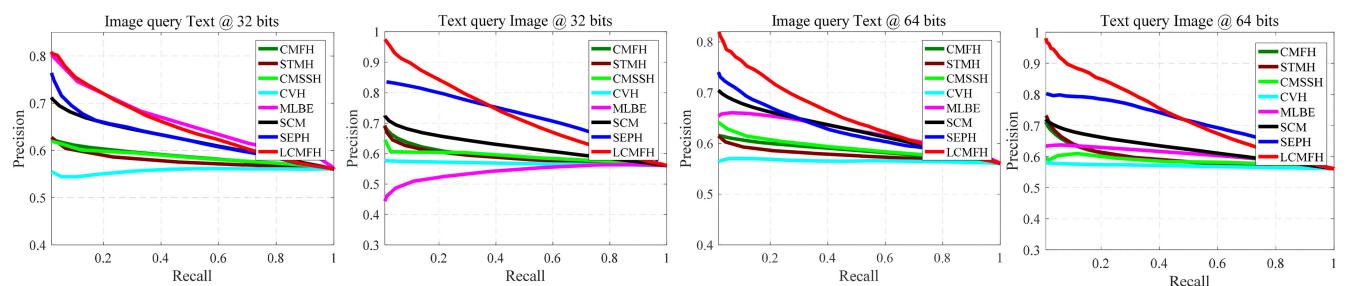


Fig. 5. Precision-Recall curves on MIRFlickr by varying code length.

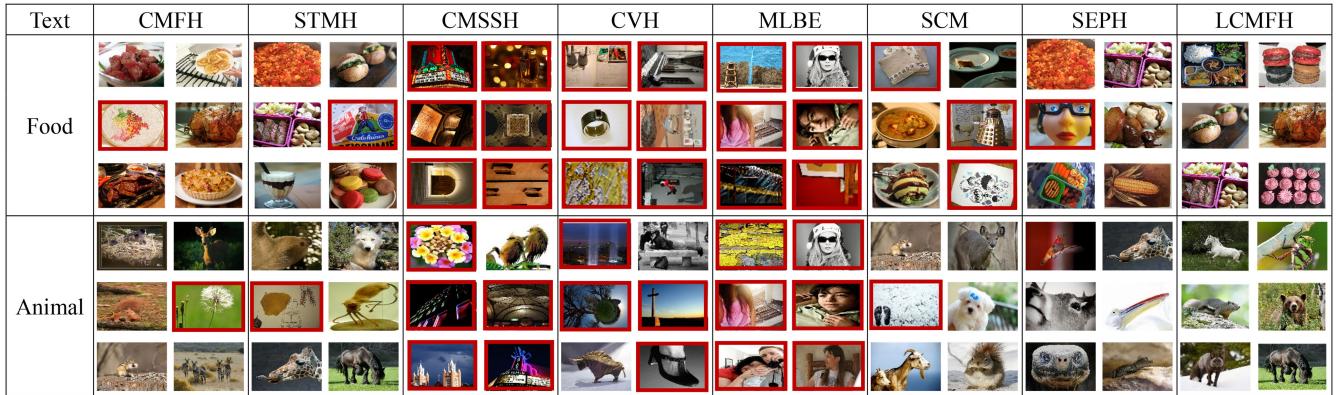


Fig. 6. Example of text-query-image retrieval on the MIRFlickr data set. Images with red border are manually marked as irrelevant.

Image	CMFH	STMH	CMSSH	CVH
	Outdoor, Day, Photo, Paix, Edmonton, Alberta, Canada, Reflection, Portrait, Me, Trinity, Window, Police, Clarify, Graffiti, Flickers, Fotografos, Spain, Heilgengamm, Demonstration, Gala, Eddy, Terrassa, Barcelona, Catalunya, Electron, Rostock, Riot, Peace, Pace.	Barcelona, City, Streets, Rime, Ponk, Tag, Bomb, Throwup, Shutters, Time, Lapse, Ride, Work, Cycling, Bicycle, Blog, Melbourne, London, Israel, Architecture, View, Eastend, Ontario, Parliament, Moose, Toy, Street, Cool, Vacation, Awesome.	Europe, Mac, England, Italy, Hills, UK, Cotswolds, Sea, War, Plains, Sky, Travel, Interestingness, Frontpage, People, Medonalds, Sargans, Clouds, Mountains, Switzerland, Apple, Gothic, Door, Globalization, Military, Bunker, Canon, Architecture, Entrata, Siracusa.	Streetart, Stencil, Edmonton, Alberta, Canada, Text, Feminism, Guitar, Snow, Man, Bay, Toronto, Storm, Cold, Snowy, Street, White, Ontario, Art, Graffiti, City, Roof, Poppies, Germany, Deutschland, Berlin, Urban, Yellow, Flowers, Orange.
	MLBE	SCM	SEPH	LCMFH
	Architecture, Rom, Crystal, Museum, Toronto, Royal, Ice, Ontario, Canada, Horses, Canberra, Act, Interestingness, Australia, Nature, Landscape, Water, Lake, Canada, Dreamscape, Photo, Winter, Beaver, Cold, Snow, Clouds, Outdoor, Frost, Buildings, Harbour.	Silhouette, Tunnel, People, Ice, Michigan, Snow, Graffiti, Cement, Urban, Italy, Interestingness, Schiphol, Blue, Hostess, Airhostess, Photo, Netherlands, Holland, Dutch, Google, Flickr, New, City, Street, Daily, Life, Diary, Europe, Airport, Travel.	Metro, Paris, France, Europe, Parisien, Chatelet, Leshalles, Man, Ledefrance, City, People, Subway, Panasonic, Rer, Icon, Street, Dog, Lumix, Interrail, Interestingness, Gothenburg, Hireen, Pedestrian, Family, Sweden, Altered, Improved, Stencil, Portugal, Shadow.	People, Commons, Explore, Iron, Bridge, Architecture, Attar, Thumb, Islamic, Clouds, Dicotomia, Cage, Man, Jump, Uomo, Skate, Salto, Wall, Wheel, Macba, Barcelona, Spain, Italy, Thailand, Temple, Isolated, River, Moto, Florence, City.

Fig. 7. Example of image-query-text retrieval on the MIRFlickr data set. Tags in blue is manually marked as relevant.

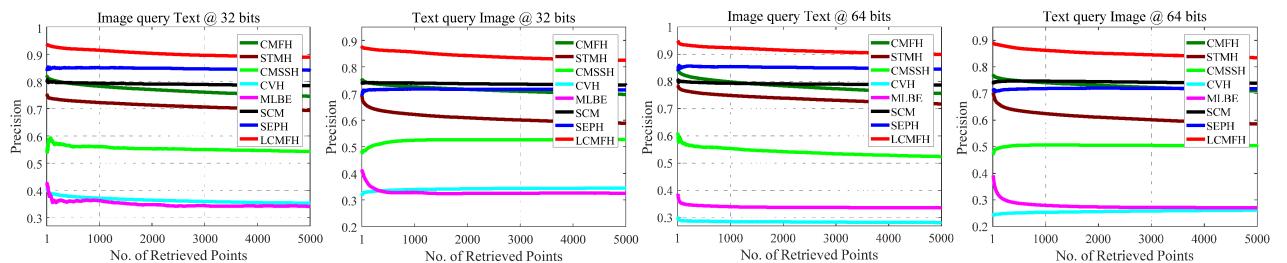


Fig. 8. topN-Precision curves on NUS-WIDE by varying code length.

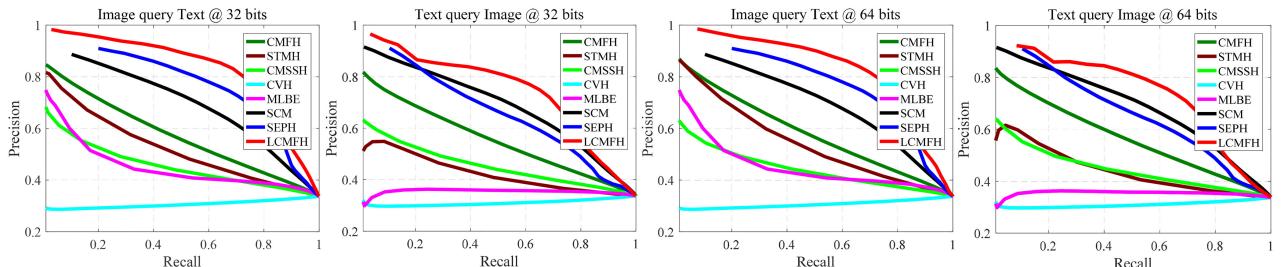


Fig. 9. Precision-Recall curves on NUS-WIDE by varying code length.

precision curves, and precision-recall curves are plotted in Figs. 8 and 9, respectively.

According to the experimental results, we can find that LCMFH still consistently outperforms the other methods. This is because none of the reference methods preserve the

discriminative information of each category during hash code learning process. In contrast, LCMFH makes full use of the semantic labels to preserve the discriminative information of each category for hash codes by guaranteeing that data points sharing the same label have the same

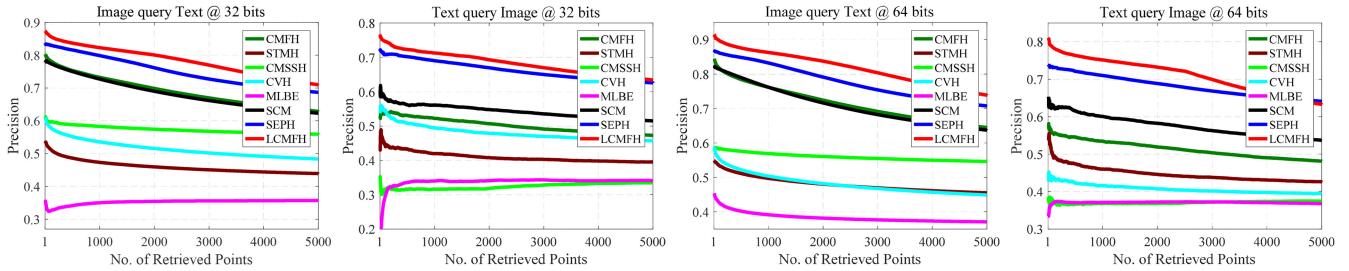


Fig. 10. topN-Precision curves on MSCOCO by varying code length.

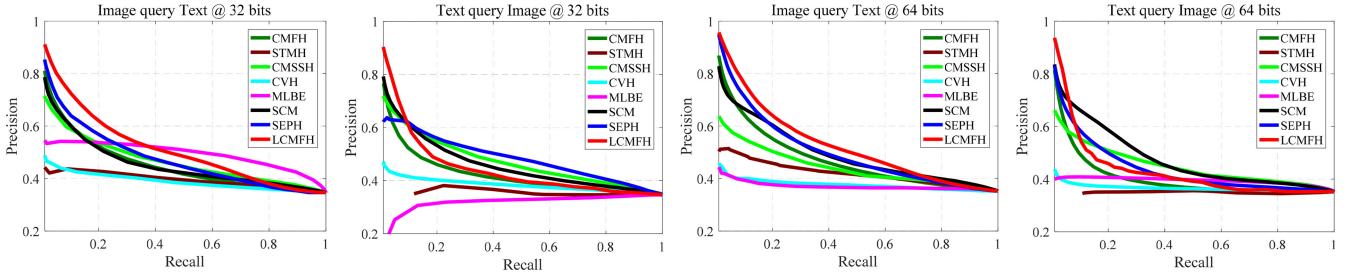


Fig. 11. Precision-Recall curves on MSCOCO by varying code length.

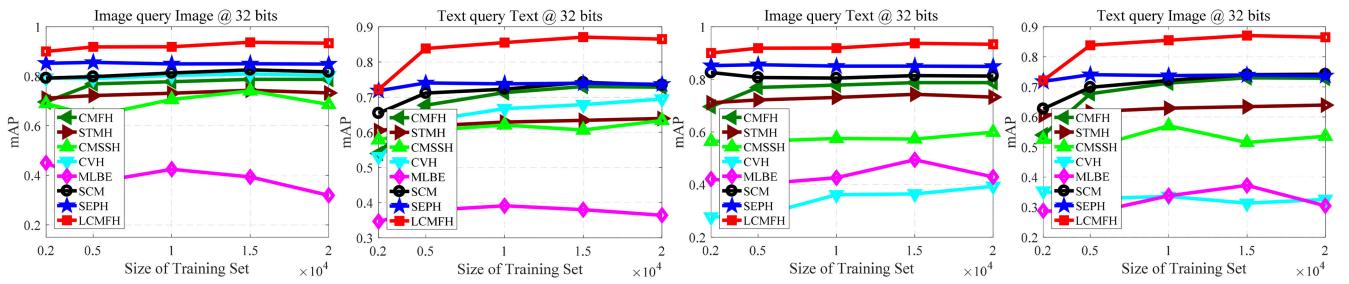


Fig. 12. mAP on NUS-WIDE by varying the size of training set.

representations. In this way, the learned hash codes by LCMFH have more discriminative power and, hence, achieve a better retrieval performance.

The experimental settings for NUS-WIDE are quite similar to the real-world scenario. The superior performance of LCMFH on the NUS-WIDE data set demonstrates that LCMFH can handle real-world large-scale multimodal similarity search problems.

### 3.5.4 Results on MSCOCO

The results on the MSCOCO data set are shown in Table 2 and Figs. 10 and 11. Consistent with the results on MIR-Flickr and NUS-WIDE, LCMFH performs better than all the baseline methods with observable margin.

By the above experiments, we can observe that LCMFH can achieve better performance than other baselines. In fact, the goal of cross-modal similarity search is to obtain semantically related instances. Therefore, by encouraging label consistency on the hash codes, LCMFH are able to exploit the semantic relationship between instances better and help build connections between modalities, which can bring better retrieval performance.

## 3.6 Effect of Training Size

In this experiment, the effect of training size on search quality is studied on the NUS-WIDE data set. Fig. 12

shows the mAP results by varying the size of the training set from 2,000 to 20,000. Clearly, the larger the training set is, the more effective the obtained hash codes are for all the methods except MLBE. The reason for the exception with MLBE is because, the training cost increases as the available training set size increases. Therefore, it is difficult for the optimizing function of MLBE to obtain the optimal value. According to the experimental results, we can find that LCMFH achieves a much better performance even on small scale training sets. This implies that LCMFH can fully utilize the available information to improve the search accuracy.

## 3.7 Convergence Study

Because LCMFH is solved by iterative update rules, the speed of convergence is important for its efficiency. In the previous sections, we have proven the convergence of the proposed updating rules. Here, we experimentally show the convergence property of LCMFH. Fig. 13 shows the convergence curves of LCMFH on all the data sets. For each figure, the  $y$ -axis is the value of the objective function and the  $x$ -axis is the iteration number. It can be observed that the update rules of LCMFH converge very quickly on all the data sets, usually within 20 iterations, which validates the effectiveness of LCMFH on large-scale data sets.

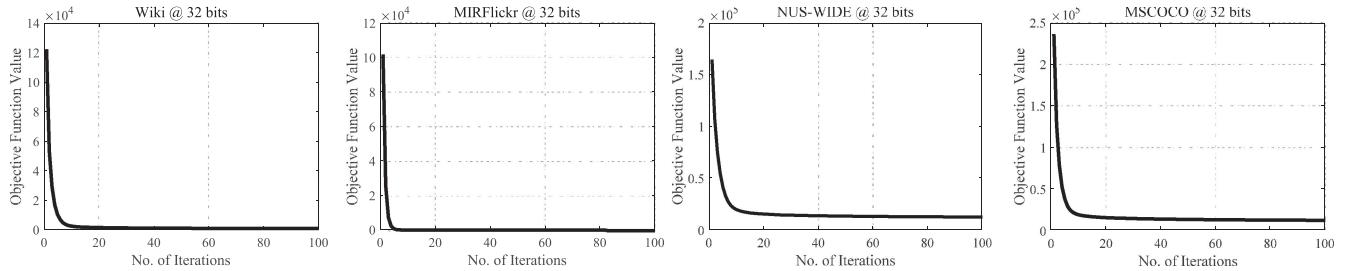


Fig. 13. Convergence curves.

### 3.8 Running Time Comparison

Table 3 shows the training time of different methods on NUS-WIDE by varying the size of the training set from 1,500 to 10,000. The code length is set to 32 in this experiment. As shown in Table 3, SEPH requires the most time for training. CMSSH and MLBE are faster than SEPH but much slower than the other methods. Because hashing is mainly designed for large-scale similarity search problems, it is obvious that SEPH, CMSSH and MLBE are not scalable due to their large training costs. CVH, CMFH, SCM, STMH, and LCMFH have approximately similar training times. However, the proposed LCMFH achieves much better performance than the others. Therefore, LCMFH possesses a competitive computational speed as well as better performance compared with existing multimodal hashing methods.

### 3.9 Parameter Sensitivity Analysis

In the previous experiments, we simply set  $\lambda = 0.5$ ,  $\mu = 10$ , and  $\gamma = 0.001$ . In this section, the empirical analysis on parameter sensitivity is given. The analysis is conducted for one parameter by varying its value while fixing the other parameters. The length of hash code is set to 32 bits. The mAP values are calculated by 10-fold cross-validation. Specifically, each data set is randomly partitioned into 10 equally sized

subsets. Next, we perform cross-modal retrieval by LCMFH, using a single subset as the query set and the remaining 9 subsets as the training set each time, and measure the corresponding retrieval performance with mAP. Finally, 10 mAP values are averaged to obtain the final results.

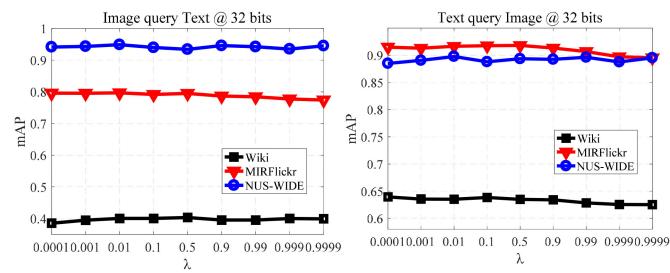
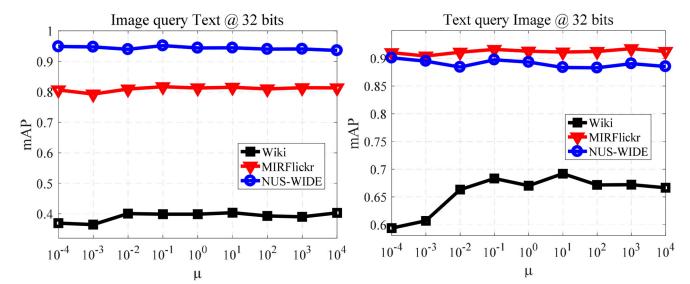
$\lambda$  is the parameter that leverages the importance of each modality. When  $\lambda$  is larger than 0.5, images have more influence on the performance of LCMFH than texts do, and vice versa. Fig. 14 illustrates the effects of  $\lambda$  on three data sets and shows that LCMFH achieves a better performance when  $\lambda$  is less than 0.5. The mAP performance decreases slightly as  $\lambda$  varies from 0.5 to 0.9999. This shows that texts have more influence than images overall on the performance of LCMFH. A possible reason is that the text features are more effective than the image features because the high-level semantic features hidden in images are more difficult to extract than those in texts. Generally,  $\lambda$  can be chosen to be within the range of [0.1, 0.6].

$\mu$  controls the influences of the label consistent matrix factorization term and the hash function learning term on the main objective function. For LCMFH, when  $\mu$  is too large, it will reduce the effectiveness of the label consistent matrix factorization. If it is too small, it will reduce the effectiveness of the hash functions. It can be found from Fig. 15 that as  $\mu$  increases, the mAP values of LCMFH first increase and then tend to be stable on the Wiki data set, while the mAP values remain stable on the MIRFlickr and NUS-WIDE data sets. This experimental results are consistent with the analysis. Fortunately, it is not difficult to choose a proper value for  $\mu$  because the performance of LCMFH is superior and steady when  $\mu \in [0.1, 1000]$ .

$\gamma$  controls the complexity of the model. The model is over-fitted when its value is too small and under-fitted when its value is too large. It can be observed from Fig. 16 that LCMFH achieves a stable performance when  $\gamma$  is less than 1. The performance of LCMFH decreases quickly when  $\gamma$  is larger than 1. Usually,  $\gamma$  can be chosen from the range between [0.00001, 0.1].

TABLE 3  
Training Time (in Seconds) on the NUS-WIDE Data Set by Varying the Size of Training Set

Method \ Size of data set	1,500	2,000	3,000	5,000	10,000
CMFH	3	4	5	7	11
STMH	4	5	7	10	19
CMSSH	314	346	367	425	1,487
CVH	12	7	5	8	23
MLBE	648	773	1,032	1,536	2,788
SCM	4	4	4	4	5
SEPH	486	654	1,081	2,306	7,495
LCMFH	3	4	5	7	12

Fig. 14. mAP versus parameter  $\lambda$ .Fig. 15. mAP versus parameter  $\mu$ .

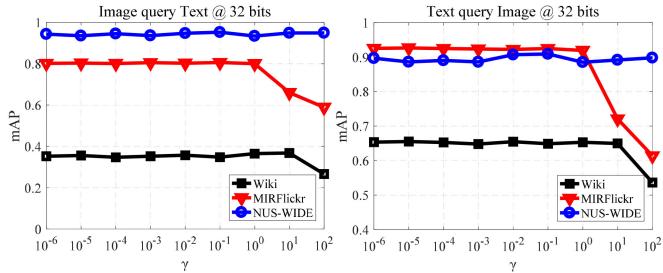


Fig. 16. mAP versus parameter  $\gamma$ .

In general, the above results demonstrate that LCMFH is sufficiently robust against the parameters.

## 4 CONCLUSIONS

In this paper, we propose a label consistent matrix factorization hashing method for large-scale cross-modal similarity searches. This method explicitly utilizes heterogeneous data along with their semantic labels to learn latent semantic spaces and unified representations, so that data points with the same labels share the same representation in the semantic spaces. Therefore, hash codes quantified by the unified representations are consistent with the semantic labels and preserve the semantic similarities. Furthermore, linear hash functions are learned for out-of-sample extensions. Extensive experiments on four benchmark data sets demonstrate the superior performance of the proposed method.

In future work, we will use a reranking method to re-rank the retrieval results of the proposed multimodal hashing method to obtain more relevant results from the query.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grants 61432014, 61702394, 61876146, 61501349, 61772402, U1605252, 61671339, and 61472304, in part by the National Key Research and Development Program of China under Grant 2016QY01W0200, in part by National High-Level Talents Special Support Program of China under Grant CS1117200001, in part by China Postdoctoral Science Foundation funded project under Grants 2018T111021, and 2017M613082, and in part by Key Industrial Innovation Chain in Industrial Domain under Grant 2016KTZDGY04-02.

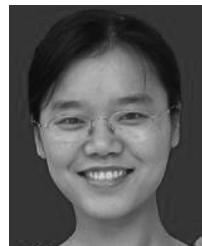
## REFERENCES

- [1] T. Takahashi and T. Kurita, "Mixture of subspaces image representation and compact coding for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1469–1479, Jul. 2015.
- [2] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—A survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.
- [3] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geom.*, 2004, pp. 253–262.
- [4] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. 12th IEEE Conf. Comput. Vis.*, 2009, pp. 2130–2137.
- [5] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.
- [6] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
- [7] X. Liu, C. Deng, B. Lang, D. Tao, and X. Li, "Query-adaptive reciprocal hash tables for nearest neighbor search," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 907–919, Feb. 2016.
- [8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.
- [9] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 1–8.
- [10] Y. Li, W. Liu, and J. Huang, "Sub-selective quantization for learning binary code in large-scale image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1526–1532, Jun. 2018.
- [11] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.
- [12] F. Shen, C. Shen, Q. Shi, A. Van Den Hengel, and Z. Tang, "Inductive hashing on manifolds," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1562–1569.
- [13] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. 26th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2938–2945.
- [14] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with Kernels," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2074–2081.
- [15] 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.
- [16] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, "Shared predictive cross-modal deep quantization," *IEEE Trans. Neural Netw. Learn. Syst.*, 2018, doi: 10.1109/TNNLS.2018.2793863.
- [17] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1618–1625.
- [18] 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*, 2013, pp. 143–152.
- [19] 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. 19th ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 785–796.
- [20] F. Zheng, Y. Tang, and L. Shao, "Hetero-manifold regularisation for cross-modal hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1059–1071, May 2018.
- [21] 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. Retrieval*, 2014, pp. 415–424.
- [22] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2083–2090.
- [23] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang, "Sparse multi-modal hashing," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 427–439, Feb. 2014.
- [24] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multi-modal hashing for cross-media retrieval," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3890–3896.
- [25] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3594–3601.
- [26] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1360–1367.
- [27] Y. Zhen and D.-Y. Yeung, "A probabilistic model for multimodal hash function learning," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 940–948.
- [28] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. 28th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3864–3872.
- [29] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.
- [30] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. 18th Int. Conf. Multimedia*, 2010, pp. 251–260.

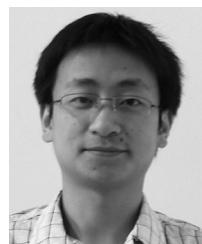
- [31] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained non-negative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.
- [32] J. C. Caicedo and F. A. González, "Multimodal fusion for image retrieval using matrix factorization," in *Proc. 2nd ACM Int. Conf. Multimedia Retrieval*, 2012, pp. 56–63.
- [33] G. Irie, H. Arai, and Y. Taniguchi, "Alternating co-quantization for cross-modal hashing," in *Proc. 15th IEEE Conf. Comput. Vis.*, 2015, pp. 1886–1894.
- [34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2014.
- [36] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, vol. 463. New York, NY, USA: ACM Press, 1999.



**Di Wang** received the PhD degree in intelligent information processing from Xidian University, Xi'an, China, in 2016. She is currently a lecturer with the School of Computer Science and Technology, Xidian University. Her research interests include machine learning and multimedia information retrieval.



**Xiumei Wang** received the PhD degree from Xidian University, in 2010. She is currently an associate professor with the School of Electronic Engineering, Xidian University. Her research interests mainly involve nonparametric statistical models and machine learning. In these areas, she has published several scientific articles including the *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, the *Pattern Recognition* and the *Neurocomputing*.



**Lihuo He** received the BSc degree in electronic and information engineering and PhD degree in pattern recognition and intelligent systems from Xidian University, Xi'an, China, in 2008 and 2013. He is currently an associate professor with Xidian University. His research interests focus on computational vision, pattern recognition and artificial intelligence. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).



**Xinbo Gao** (M'02-SM'07) received the BEng, MSc, and PhD degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively. From 1997 to 1998, he was a research fellow with the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a post-doctoral research fellow with the Department of Information Engineering, the Chinese University of Hong Kong, Hong Kong. Since 2001, he has been at the School of Electronic Engineering, Xidian University. He is currently a Cheung Kong professor of Ministry of Education, a professor of Pattern Recognition and Intelligent System, and the director of the State Key Laboratory of Integrated Services Networks, Xi'an, China. His current research interests include multimedia analysis, computer vision, pattern recognition, machine learning, and wireless communications. He has published six books and around 200 technical articles in refereed journals and proceedings. He is on the Editorial Boards of several Journals, including the *Signal Processing* (Elsevier) and the *Neurocomputing* (Elsevier). He served as the general chair/co-chair, program committee chair/co-chair, or PC Member for around 30 major international conferences. He is a fellow of the Institute of Engineering and Technology and a fellow of the Chinese Institute of Electronics. He is a senior member of the IEEE.