Fast Cross-Modal Hashing With Global and Local Similarity Embedding

Yongxin Wang, Zhen-Duo Chen, Xin Luo, Rui Li, and Xin-Shun Xu, *Member, IEEE*

***Abstract*—Recently, supervised cross-modal hashing has attracted much attention and achieved promising performance. To learn hash functions and binary codes, most methods globally exploit the supervised information, for example, preserving an at-least-one pairwise similarity into hash codes or reconstructing the label matrix with binary codes. However, due to the hardness of the discrete optimization problem, they are usually time consuming on large-scale datasets. In addition, they neglect the class correlation in supervised information. From another point of view, they only explore the global similarity of data but overlook the local similarity hidden in the data distribution. To address these issues, we present an efficient supervised cross-modal hashing method, that is, fast cross-modal hashing (FCMH). It leverages not only global similarity information but also the local similarity in a group. Specifically, training samples are partitioned into groups; thereafter, the local similarity in each group is extracted. Moreover, the class correlation in labels is also exploited and embedded into the learning of binary codes. In addition, to solve the discrete optimization problem, we further propose an efficient discrete optimization algorithm with a well-designed group updating scheme, making its computational complexity linear to the size of the training set. In light of this, it is more efficient and scalable to large-scale datasets. Extensive experiments on three benchmark datasets demonstrate that FCMH outperforms some state-of-the-art cross-modal hashing approaches in terms of both retrieval accuracy and learning efficiency.**

***Index Terms*—Cross-modal hashing, discrete optimization, local similarity embedding, scalable hashing.**

I. Introduction

With the rapid growth of multimedia data on the Internet, the general nearest-neighbor search becomes impractical for large-scale retrieval tasks because of its high time and space consumption. Therefore, the approximate nearest neighbor (ANN) search is proposed with the aim of seeking a balance between accuracy and efficiency. As a result, hashing [1]–[3], as a representative ANN search technique, has attracted much attention in recent years. It compresses high-dimensional features into binary hash codes while preserving the similarity of data; thereafter, the search can be performed efficiently in the Hamming space with the XOR operation, significantly reducing time and space costs.

More recently, there has been an increasing demand for cross-modal retrieval [4]. For example, given a query with one modality, for example, text, users usually hope to obtain relevant instances with other modalities, for example, images. Consequently, many cross-modal hashing methods have been proposed and achieved promising performance. According to whether supervised information is used, existing cross-modal hashing methods can be classified into unsupervised and supervised ones. Generally speaking, supervised ones can generate more discriminative hash codes and achieve better retrieval performance than unsupervised ones. However, there are still several essential issues to be further considered.

1) How to fully exploit the supervised information in data? For example, some hashing models [5]–[7] construct an at-least-one pairwise similarity matrix and approximate it by the symmetric inner product of to-be-learned hash codes. The similarity matrix is usually an $n \times n$ matrix, leading to $O(n^2)$ time and space complexity. In addition, this binary similarity cannot exactly reflect the semantic relationship between samples. Therefore, some methods [8]–[11] proposed to reconstruct labels with to-be-learned hash codes, which is much more efficient. Nevertheless, they still ignore the useful class correlation in labels. More important, they only explore the global similarity of heterogeneous data but neglect the importance of local similarity within a local group, making the retrieval results less fine grained.
2) How to efficiently solve the discrete optimization problem? Due to the hardness of discrete optimization problems, some methods [10], [12] relax the binary constraint and employ a relaxation strategy to generate approximate continuous values, leading to large quantization error and ineffective hash codes. To tackle this, some methods solve the discrete problem via a bit-by-bit optimization scheme [13], [14], which is inefficient and unscalable to large-scale datasets.

To address the above issues, in this article, we present a novel supervised cross-modal hashing method, namely, fast

Manuscript received June 18, 2020; revised November 1, 2020; accepted February 5, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61872428; in part by the Shandong Provincial Key Research and Development Program under Grant 2019JZZY010127; in part by the Natural Science Foundation of Shandong Province under Grant ZR2019ZD06; in part the by the Major Program of the National Natural Science Foundation of China under Grant 61991411; in part by the Natural Science Foundation of Shandong Province under Grant ZR2020QF036; and in part by the Fundamental Research Funds of Shandong University under Grant 019GN075. This article was recommended by Associate Editor H. Liu. *(Corresponding author: Xin-Shun Xu.)*

Yongxin Wang, Zhen-Duo Chen, Xin Luo, and Xin-Shun Xu are with the School of Software, Shandong University, Jinan 250101, China (e-mail: yxinwang@hotmail.com; chenzd.sdu@gmail.com; luoxin.lxin@gmail.com; xuxinshun@sdu.edu.cn).

Rui Li is with the AI Research Institute, Inspur Inc., Jinan 250101, China (e-mail: lirui01@inspur.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3059886.

Digital Object Identifier 10.1109/TCYB.2021.3059886

cross-modal hashing (FCMH). Specifically, it exploits supervised information through pairwise similarity preserving and correlated class reconstructing, which not only avoids using the $n \times n$ similarity matrix but also takes the class correlation into consideration. Therefore, it is much more efficient and effective. To further explore the local structure of data, it divides samples into groups according to data distribution and extracts the local similarity in each group. Thereafter, both the global and local similarities are embedded in the learning of binary codes through a well-designed group updating scheme, leading to more effective hash codes. Furthermore, FCMH discretely generates hash codes without relaxation; therefore, the quantization error can be further reduced. Overall, FCMH seeks better retrieval performance, including retrieval accuracy and learning efficiency. Considering this, FCMH avoids the limitations of the existing methods mentioned previously. The main contributions of this article are summarized as follows.

1) We present an FCMH method. It not only fully exploits the supervised information by pairwise similarity preserving and correlated label reconstructing but also takes both global and local similarities of data into consideration. Thus, FCMH is able to generate more discriminative hash codes, resulting in more fine-grained retrieval results.
2) An efficient discrete optimization algorithm is proposed to solve the discrete problem with no relaxation, avoiding the quantization error problem. In addition, its time and space complexity is linear to the size of the training set, making it scalable to large-scale datasets. In particular, we design a group updating scheme, which seamlessly integrates both global and local similarities into a unified optimization framework. Moreover, the group updating scheme can be performed in parallel, which will further accelerate the optimization speed.
3) Extensive experiments are conducted on three widely used datasets, and the results verify that FCMH outperforms several state-of-the-art cross-modal hashing methods, in terms of both accuracy and efficiency.

The remainder of the article is organized as follows. Section II briefly reviews some related works. Section III gives the details of FCMH. Section IV presents the experimental results and discussions, followed by the conclusion in Section V.

II. Related Work

In recent years, many hashing methods have been proposed and attracted much attention [15]–[17]. Typical examples include, but are not limited to, iterative quantization (ITQ) [18], discrete graph hashing (DGH) [14], neighborhood discriminant hashing (NDH) [19], discrete locally linear embedding hashing (DLLH) [20], binary multidimensional scaling (BMDS) [21], supervised hashing with kernels (KSH) [22], supervised discrete hashing (SDH) [13], fast supervised discrete hashing (FSDH) [23], fast scalable supervised hashing (FSSH) [24], and discrete hashing with multiple supervision (MSDH) [25]. These models are designed to learn binary codes for unimodal data, which cannot deal with multiple types of features from multimodal data.

To conduct cross-modal retrieval, many cross-modal hashing methods have been proposed, which can be classified into unsupervised and supervised ones. Unsupervised ones explore the intrinsic correlation of multiple modalities from original data and embed the obtained correlation into binary codes. For example, collective matrix factorization hashing (CMFH) [26] is the first work to jointly learn the latent factor of different modalities by collective matrix factorization. Fusion similarity hashing (FSH) [27] constructs a fused similarity graph of multiple modalities and then preserves the similarity into the Hamming space. Robust and flexible discrete hashing (RFDH) [28] generates binary codes by discrete collective matrix decomposition and learns hash functions via a classification model. Collective reconstructive embedding (CRE) [29] addresses the heterogeneity problem by applying a domain-specific distance for different modalities. However, these unsupervised methods cannot exploit the given semantic information, leading to large semantic gaps.

In contrast, supervised cross-modal hashing further leverages supervised information to generate more discriminative hash codes, which have exhibited better performance than unsupervised ones. Consequently, it has attracted increasing attention in recent years. According to the form of semantics, it can be further classified into pairwise-similarity-based and label-based ones. The former construct the pairwise similarity matrix and optimize an inner product minimization problem. As a result, semirelaxation supervised hashing (SRSH) [6] factorizes a pairwise similarity matrix and simplifies the NP-hard optimization by relaxing a part of binary constraints. Semantics preserving hashing (SePH) [5] transforms the semantic similarity into a probability distribution and embeds it into hash codes by minimizing the Kullback–Leibler divergence. Generalized semantic preserving hashing (GSPH) [7] first embeds the semantic similarity into hash codes via a relaxation scheme and then learns a projection matrix to map original data into the Hamming space. Discrete latent factor hashing (DLFH) [30] designs a discrete latent factor framework to model the semantic information into hash codes. Generally speaking, due to the $n \times n$ pairwise similarity matrix, these methods are usually space and time consuming. In contrast, the latter directly leverages the label matrix and solves a binary classification problem, making their space and time complexity linear to the size of the training set. For instance, discriminative cross-modal hashing (DCH) [8] discretely learns hash codes and hash functions by a linear regression framework under the supervision of labels. Fast discrete cross-modal hashing (FDCH) [9] regards the labels as high-level features and reconstructs binary codes by labels with a draft. Label consistent matrix factorization hashing (LCMFH) [10] directly leverages labels and adopts matrix factorization to learn a latent semantic space of heterogeneous data. Scalable discrete matrix factorization hashing (SCRATCH) [31] exploits labels and kernel features to find a latent semantic space. Subspace relation learning for cross-modal hashing (SRLCH) [11] transforms labels to subspace relation information; meanwhile, it jointly learns the unified

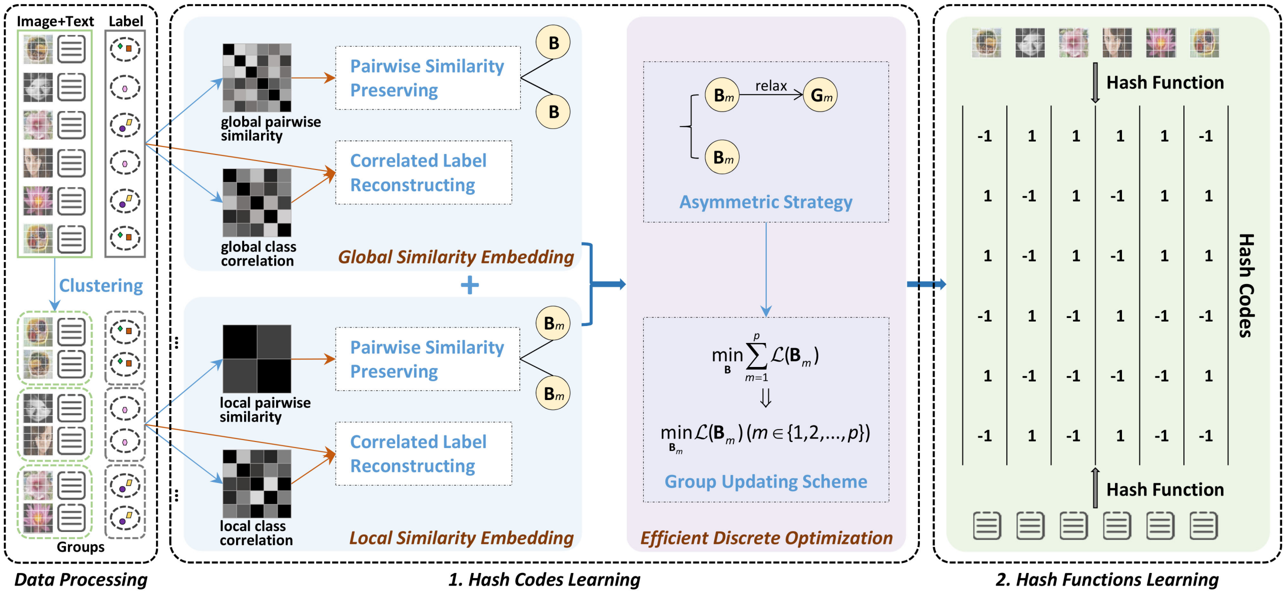

Image+Text
Label
Clustering
Groups
global pairwise similarity
global class correlation
Pairwise Similarity Preserving
Correlated Label Reconstructing
B
B
Global Similarity Embedding
+
local pairwise similarity
local class correlation
Pairwise Similarity Preserving
Correlated Label Reconstructing
B_m
B_m
Local Similarity Embedding
B
B_m
relax
G_m
B_m
Asymmetric Strategy
$\min_{B} \sum_{m=1}^{p} \mathcal{L}(B_m)$
⇓
$\min_{B_m} \mathcal{L}(B_m)\, (m \in \{1,2,\ldots,p\})$
Group Updating Scheme
Efficient Discrete Optimization
Hash Function
-1 1 1 1 1 -1
1 -1 1 -1 1 1
1 -1 1 -1 1 1
-1 1 -1 1 -1 -1
1 -1 -1 -1 -1 1
-1 1 -1 1 -1 -1
Hash Codes
Hash Function
Data Processing
1. Hash Codes Learning
2. Hash Functions Learning

Fig. 1. Workflow of FCMH.

binary codes and hash functions by minimizing the distance between relation information and hash codes. However, all aforementioned supervised cross-modal hashing methods ignore the class correlation in labels. More important, none of them explores the local structure hidden in original data. We argue that the class correlation and local structure are valuable in generating fine-grained retrieval results.

Inspired by the success of deep learning, several deep hashing methods [32]–[35] have been proposed. Compared with shallow methods, they can directly deal with raw data rather than handcrafted features. However, most of them are time consuming and cannot optimize complex objective functions. Therefore, in this article, we mainly focus on how to generate effective hash codes and how to efficiently solve binary optimization problems.

Different from the existing cross-modal hashing methods, in this article, FCMH focuses on how to fully exploit the information of data to achieve more fine-grained retrieval accuracy and higher learning efficiency.

III. Method

A. Notations and Framework

In this article, we use boldface uppercase letters to denote matrices, for example, $\mathbf{X}$. Correspondingly, $\mathbf{X}_{i*}$, $\mathbf{X}_{*j}$, and $\mathbf{X}_{ij}$ denote the ith row, jth column, and (i, j)th element of $\mathbf{X}$, respectively. $\mathbf{X}_m$ indicates a submatrix of $\mathbf{X}$. $\mathbf{I}$, $\mathbf{1}$, and $\mathbf{0}$ indicates an identify matrix, an all one vector, and an all-zero vector, respectively. $\mathrm{tr}(\cdot)$ means the trace operator. $\mathrm{sign}(\cdot)$ is the sign function which returns 1 for non-negative inputs and -1 for negative inputs. $\|\cdot\|$ denotes the 2-norm for a vector and the Frobenius-norm for a matrix. $[\cdot]$ is a pair of separators.

Assume that $\mathbf{X}^{(k)} \in \Re^{d_k \times n} (k \in \{1, 2, \ldots, o\})$ is the training set of the kth modality, where n is the number of instances, o is the number of modalities, and d_k is the feature dimensionality. $\mathbf{Y} \in \Re^{c \times n}$ is the label matrix, where c is the number of categories. $\mathbf{Y}_{ij} = 1$ if the jth sample belongs to the ith category and 0, otherwise. $\mathbf{B} \in \{-1, 1\}^{r \times n}$ is the to-be-learned unified hash codes, where r denotes the hash code length.

The framework of FCMH is illustrated in Fig. 1, which is composed of three parts, that is: 1) data processing; 2) hash codes learning; and 3) hash functions learning. More specifically, during data processing, it clusters the training data of multiple modalities into several groups via multiview clustering. For hash codes learning, it first leverages labels to construct the global and local similarities and then embeds them into hash codes by preserving global and local similarity. Subsequently, it solves the discrete optimization problem via an asymmetric strategy and a well-designed group updating scheme. Finally, hash functions are learned under the supervision of the generated hash codes. It is worth noting that FCMH adopts a two-step strategy for hash learning. In other words, the hash learning process is divided into two steps: 1) hash codes learning and 2) hash functions learning. Unlike one-step methods that learn hash codes and hash functions simultaneously, the two-step strategy is able to reduce the complication of optimization and avoids hash codes and hash functions interfering with each other. The details of these components are described in the following sections.

B. Hash Codes Learning

To generate effective hash codes, we take both global and local similarities of data into consideration. In addition, an efficient optimization algorithm is designed to discretely generate binary codes without relaxation.

1) Global Similarity Embedding: In the existing supervised cross-modal hashing methods, generally speaking, there are two schemes to leverage supervised information to learn binary codes or hash functions: 1) constructing a pairwise similarity matrix of all samples in training set [6], [7], [22] and 2) using a label matrix as supervised information and learning a mapping function from binary codes to the label matrix [8], [13].

For example, based on the first scheme, the learning problem can be defined as the following *pairwise similarity preserving* problem with inner product minimization:

$$\min_{\mathbf{B}} \left\| \mathbf{B}^\top \mathbf{B} - r\mathbf{S}^g \right\|^2, \text{ s.t. } \mathbf{B} \in \{-1, 1\}^{r\times n} \tag{1}$$

where $\mathbf{S}^g$ is the pairwise similarity matrix. Traditionally, $\mathbf{S}^g_{ij} = 1$ if the ith and jth points share at least one common label, and $\mathbf{S}^g_{ij} = -1$, otherwise. This binary similarity matrix is able to well capture the similarity information between samples when each sample only belongs to one category, that is, each sample has one label. However, in most real applications, one sample usually belongs to more than one category, that is, having multiple labels. In such a scenario, the degree of similarity between samples is different, and the above similarity matrix cannot reflect it, leading to the loss of semantic information. To address this, some methods [8], [13] directly use labels as supervised information to learn binary codes or hash functions by solving the following label reconstructing problem:

$$\min_{\mathbf{B},\mathbf{P}} \|\mathbf{Y} - \mathbf{P}\mathbf{B}\|^2 + \gamma\|\mathbf{P}\|^2, \text{ s.t. } \mathbf{B} \in \{-1, 1\}^{r\times n} \tag{2}$$

where γ is a penalty parameter to avoid overfitting, and $\mathbf{P} \in \Re^{c\times r}$ is a projection matrix. By optimizing (2), the label information is embedded, making the learned binary codes more discriminative.

Apparently, both of the above schemes integrally capture the similarity information of all samples; therefore, we regard them as global similarity embedding models. Actually, the first scheme can be regarded as a kind of coarse-grained method, which only measures whether two samples are similar; the second one can be viewed as a middle-grained approach, which can partly model the degree of similarity. However, there are several problems in the above schemes. First, the size of the pairwise similarity matrix is $n \times n$. Therefore, if no special computational trick is used, the complexity of time and storage during optimization may be $O(n^2)$ at least, making the hash learning process inefficient on large-scale datasets. Second, both of them cannot exploit the class correlation in supervised information, which has been widely explored in multilabel learning [36]–[38]. To overcome these issues, inspired by [12], we first redefine the *global pairwise similarity* matrix as follows:

$$\mathbf{S}^g = 2\tilde{\mathbf{Y}}^\top\tilde{\mathbf{Y}} - \mathbf{1}\mathbf{1}^\top \tag{3}$$

where $\tilde{\mathbf{Y}}$ is a 2-norm column normalized label matrix, with its jth column defined as $\tilde{\mathbf{Y}}_{*j} = \mathbf{Y}_{*j}/\|\mathbf{Y}_{*j}\|$. Obviously, $\mathbf{S}^g \in [-1, 1]$ is symmetric, and it can carry more semantic information than the binary similarity matrix, especially for multilabel data. Moreover, by directly using the right-hand side of (3) in calculations, the $n \times n$ similarity matrix can be avoided; therefore, time and space costs can be reduced.

To exploit the class correlation in supervised information, we further develop (2) into the following *correlated label reconstructing* problem:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{P}} \quad & \sum_{i,j=1}^{c} \left[\mathbf{C}^g\right]_{ij} \left\|\mathbf{Y}_{i*} - [\mathbf{P}\mathbf{B}]_{j*}\right\|^2 + \gamma\|\mathbf{P}\|^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r\times n}\end{aligned} \tag{4}$$

where $\mathbf{C}^g \in \Re^{c\times c}$ is the *global class correlation* matrix, which is defined as follows:

$$\mathbf{C}^g = \ddot{\mathbf{Y}}\ddot{\mathbf{Y}}^\top. \tag{5}$$

$\ddot{\mathbf{Y}}$ is a 2-norm row normalized label matrix, with its ith row defined as $\ddot{\mathbf{Y}}_{i*} = \mathbf{Y}_{i*}/\|\mathbf{Y}_{i*}\|$. It is obvious that $\mathbf{C}^g \in [0, 1]$ is a symmetric matrix and its diagonal elements are all one. The larger the value of $[\mathbf{C}^g]_{ij}$, the higher the frequency of labels $\mathbf{Y}_{i*}$ and $\mathbf{Y}_{j*}$ co-occurrence, that is, if an instance belongs to the ith category, it is more likely that it also belongs to the jth category. By further considering the class correlation, the binary codes are encouraged to be similar on highly positively correlated categories, and dissimilar on highly negatively correlated categories, leading to more fine-grained hash codes.

Thereafter, combining (1) and (4), we define the following objective function to embed global similarity:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{P}} \quad & \alpha\left\|\mathbf{B}^\top\mathbf{B} - r\mathbf{S}^g\right\|^2 + \gamma\|\mathbf{P}\|^2 \\ & + \beta\sum_{i,j=1}^{c}\left[\mathbf{C}^g\right]_{ij}\left\|\mathbf{Y}_{i*} - [\mathbf{P}\mathbf{B}]_{j*}\right\|^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r\times n}\end{aligned} \tag{6}$$

where α and β are tradeoff parameters and γ is a penalty parameter for regularization.

2) Local Similarity Embedding: As we know, data usually contain a local structure, which is helpful in computer vision tasks [38]. Further considering the local structure may make visually and semantically similar samples more similar, and visually similar but semantically dissimilar samples more dissimilar. Therefore, if we could extract local similarity and embed it into the learning of binary codes, it is very possible that we can obtain more fine-grained retrieval results in terms of both semantics and vision.

For this purpose, we assume the training set of the kth modality is partitioned into p groups, that is, $\mathbf{X}^{(k)} = [\mathbf{X}^{(k)}_1, \mathbf{X}^{(k)}_2, \ldots, \mathbf{X}^{(k)}_p]$, where $\mathbf{X}^{(k)}_m \in \Re^{d_k\times n_m}(m \in \{1, 2, \ldots, p\})$. This partition can be generated by knowledge or clustering. $\mathbf{Y}_m$, $\tilde{\mathbf{Y}}_m$, and $\mathbf{B}_m$ are the corresponding subgroup of $\mathbf{Y}$, $\tilde{\mathbf{Y}}$, and $\mathbf{B}$, respectively. Thereafter, similar to the global similarity matrix, that is, (3), the *local pairwise similarity* matrix of the mth group can be calculated as follows:

$$\mathbf{S}^l_m = 2\tilde{\mathbf{Y}}^\top_m\tilde{\mathbf{Y}}_m - \mathbf{1}\mathbf{1}^\top. \tag{7}$$

The same as the global class correlation, that is, (5), the *local class correlation* of $\mathbf{X}^{(k)}_m$ is defined as follows:

$$\mathbf{C}^l_m = \ddot{\mathbf{Y}}_m\ddot{\mathbf{Y}}^\top_m \tag{8}$$

where $\ddot{\mathbf{Y}}_m$ is a 2-norm row normalized matrix of $\mathbf{Y}_m$, with its ith row defined as $[\ddot{\mathbf{Y}}_m]_{i*} = [\mathbf{Y}_m]_{i*}/\|[\mathbf{Y}_m]_{i*}\|$.

Similar to (6), we define the following objective function to embed the local similarity into the learning of binary codes

$$\begin{aligned}\min_{\mathbf{B},\mathbf{P}} \quad & \alpha\sum_{m=1}^{p}\left\|\mathbf{B}^\top_m\mathbf{B}_m - r\mathbf{S}^l_m\right\|^2 + \gamma\|\mathbf{P}\|^2 \\ & + \beta\sum_{m=1}^{p}\sum_{i,j=1}^{c}\left[\mathbf{C}^l_m\right]_{ij}\left\|[\mathbf{Y}_m]_{i*} - [\mathbf{P}\mathbf{B}_m]_{j*}\right\|^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{r\times n}\end{aligned} \tag{9}$$

where α and β are tradeoff parameters and γ is a penalty parameter for regularization.

3) Overall Objective Function: Combining (6) and (9), that is, global similarity embedding and local similarity embedding, we define the following overall objective function to embed both global and local similarities:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{P}}\ & \alpha_1\left\|\mathbf{B}^\top\mathbf{B}-r\mathbf{S}^g\right\|^2\\ &+\alpha_2\sum_{m=1}^{p}\left\|\mathbf{B}_m^\top\mathbf{B}_m-r\mathbf{S}_m^l\right\|^2+\gamma\|\mathbf{P}\|^2\\ &+\beta_1\sum_{i,j=1}^{c}\left[\mathbf{C}^g\right]_{ij}\left\|\mathbf{Y}_{i*}-[\mathbf{PB}]_{j*}\right\|^2\\ &+\beta_2\sum_{m=1}^{p}\sum_{i,j=1}^{c}\left[\mathbf{C}_m^l\right]_{ij}\left\|[\mathbf{Y}_m]_{i*}-[\mathbf{PB}_m]_{j*}\right\|^2\\ \text{s.t.}\ & \mathbf{B}\in\{-1,1\}^{r\times n}\end{aligned}\tag{10}$$

where α_1, α_2, β_1, and β_2 are balance parameters to adjust the importance of different items. For ease of representation, we assume there is a common projection matrix $\mathbf{P}$ in both the global and local correlated label reconstruction. Actually, the fourth and fifth objective items in (10), corresponding to correlated label reconstructing, can be reformulated into the following form:

$$\begin{aligned}\sum_{m=1}^{p}\sum_{i,j=1}^{c}\Big(&\beta_1\left[\mathbf{C}^g\right]_{ij}\left\|[\mathbf{Y}_m]_{i*}-[\mathbf{PB}_m]_{j*}\right\|^2\\ &+\beta_2\left[\mathbf{C}_m^l\right]_{ij}\left\|[\mathbf{Y}_m]_{i*}-[\mathbf{PB}_m]_{j*}\right\|^2\Big).\end{aligned}\tag{11}$$

From this perspective, it can be regarded as mapping the to-be-learned hash codes to labels under the weighting of global and local class correlations, with β_1 and β_2 as balance parameters.

4) Efficient Discrete Optimization: However, the optimization problem of (10) is NP-hard, due to the symmetric inner product of binary codes. To solve this problem, we replace one $\mathbf{B}_m$ with a real-valued $\mathbf{G}_m$ and introduce a regularization term between them with an orthogonal rotation matrix, called *asymmetric strategy*. Moreover, to prevent $\mathbf{G}_m$ from being biased, orthogonal and equilibrium constraints [39] are further put on it. Thereafter, (10) is transformed into the following one:

$$\begin{aligned}\min_{\mathbf{P},\mathbf{G},\mathbf{R},\mathbf{B}}\ & \|\mathbf{B}-\mathbf{RG}\|^2+\gamma\|\mathbf{P}\|^2\\ &+\alpha_1\left\|\mathbf{G}^\top\mathbf{B}-r\mathbf{S}^g\right\|^2+\alpha_2\sum_{m=1}^{g}\left\|\mathbf{G}_m^\top\mathbf{B}_m-r\mathbf{S}_m^l\right\|^2\\ &+\beta_1\sum_{m=1}^{p}\sum_{i,j=1}^{c}\left[\mathbf{C}^g\right]_{ij}\left\|[\mathbf{Y}_m]_{i*}-[\mathbf{PG}_m]_{j*}\right\|^2\\ &+\beta_2\sum_{m=1}^{p}\sum_{i,j=1}^{c}\left[\mathbf{C}_m^l\right]_{ij}\left\|[\mathbf{Y}_m]_{i*}-[\mathbf{PG}_m]_{j*}\right\|^2\\ \text{s.t.}\ & \mathbf{B}_m\in\{-1,1\}^{r\times n_m},\ \mathbf{RR}^\top=\mathbf{I}\\ & \mathbf{G}_m\mathbf{G}_m^\top=n_m\mathbf{I},\ \mathbf{G}_m\mathbf{1}=\mathbf{0}\end{aligned}\tag{12}$$

where $\mathbf{R}\in\Re^{r\times r}$ is an orthogonal rotation matrix [18]. The asymmetric strategy is inspired by asymmetric hashing [40], [41], which learns different hash functions for queries and the database and has been proven effective in some recent hashing literature [42], [43]. With this strategy, it not only preserves the binary constraint but also gets rid of the symmetric binary matrix factorization, avoiding the quantization error problem and making it efficient during optimization. Moreover, it is able to achieve better accuracy because the real-valued substitution $\mathbf{G}_m$ can carry more precise correlations of data.

Due to the constraints $\mathbf{G}_m\mathbf{G}_m^\top=n_m\mathbf{I}$, $\mathbf{B}_m\in\{-1,1\}^{r\times n_m}$, and $\mathbf{RR}^\top=\mathbf{I}$, (12) can be equally transformed into the following matrix trace form:

$$\begin{aligned}\max_{\mathbf{P},\mathbf{G},\mathbf{R},\mathbf{B}}\ & 2\,\mathrm{tr}\left(\mathbf{R}^\top\mathbf{BG}^\top\right)-\gamma\,\mathrm{tr}\left(\mathbf{P}^\top\mathbf{P}\right)\\ &+2\alpha_1\,\mathrm{tr}\left(r\mathbf{BS}^g\mathbf{G}^\top\right)+\sum_{m=1}^{p}\left(2\alpha_2\,\mathrm{tr}\left(r\mathbf{B}_m\mathbf{S}_m^l\mathbf{G}_m^\top\right)\right)\\ &+\sum_{m=1}^{p}\left(\beta_1\,\mathrm{tr}\left(2\mathbf{P}^\top\mathbf{C}^g\mathbf{Y}_m\mathbf{G}_m^\top-n_m\mathbf{P}^\top\mathbf{D}^g\mathbf{P}\right)\right)\\ &+\sum_{m=1}^{p}\left(\beta_2\,\mathrm{tr}\left(2\mathbf{P}^\top\mathbf{C}_m^l\mathbf{Y}_m\mathbf{G}_m^\top-n_m\mathbf{P}^\top\mathbf{D}_m^l\mathbf{P}\right)\right)\\ \text{s.t.}\ & \mathbf{B}_m\in\{-1,1\}^{r\times n_m},\ \mathbf{RR}^\top=\mathbf{I}\\ & \mathbf{G}_m\mathbf{G}_m^\top=n_m\mathbf{I},\ \mathbf{G}_m\mathbf{1}=\mathbf{0}\end{aligned}\tag{13}$$

where $\mathbf{D}^g$ is the degree matrix of $\mathbf{C}^g$ defined as $[\mathbf{D}^g]_{ii}=\sum_{j=1}^{c}[\mathbf{C}^g]_{ij}$. Similarly, $[\mathbf{D}_m^l]_{ii}=\sum_{j=1}^{c}[\mathbf{C}_m^l]_{ij}$.

Subsequently, to solve the problem of (13), we propose an iterative optimization algorithm to discretely generate binary codes. All steps are sequentially repeated until convergence. Specifically, all variables are alternatively updated with other variables fixed, and $\mathbf{B}$ and $\mathbf{G}$ are updated by a well-designed group updating scheme. The details are described as follows.

***P**-Step:* By setting the derivative of (13) with respect to $\mathbf{P}$ to 0, we have the solution of $\mathbf{P}$ as follows:

$$\begin{aligned}\mathbf{P}=&\left(\gamma\mathbf{I}+\sum_{m=1}^{p}\left(n_m\left(\beta_1\mathbf{D}^g+\beta_2\mathbf{D}_m^l\right)\right)\right)^{-1}\\ &\times\sum_{m=1}^{p}\left(\left(\beta_1\mathbf{C}^g+\beta_2\mathbf{C}_m^l\right)\mathbf{Y}_m\mathbf{G}_m^\top\right).\end{aligned}\tag{14}$$

***G**-Step:* With $\mathbf{P}$, $\mathbf{B}$, and $\mathbf{R}$ fixed, (13) can be rewritten into the following subproblem:

$$\begin{aligned}\max_{\mathbf{G}}\ & \sum_{m=1}^{p}\mathrm{tr}\Big(\Big(\mathbf{R}^\top\mathbf{B}_m+\alpha_1 r\mathbf{BS}_m^g+\alpha_2 r\mathbf{B}_m\mathbf{S}_m^l\\ &\qquad+\mathbf{P}^\top\left(\beta_1\mathbf{C}^g+\beta_2\mathbf{C}_m^l\right)\mathbf{Y}_m\Big)\mathbf{G}_m^\top\Big)\\ \text{s.t.}\ & \mathbf{G}_m\mathbf{G}_m^\top=n_m\mathbf{I},\ \mathbf{G}_m\mathbf{1}=\mathbf{0}\end{aligned}\tag{15}$$

where $\mathbf{S}_m^g\in[-1,1]^{n\times n_m}$ is the submatrix of $\mathbf{S}^g$, defined as $\mathbf{S}_m^g=\tilde{\mathbf{Y}}^\top\tilde{\mathbf{Y}}_m-\mathbf{1}\mathbf{1}^\top$. Obviously, as defined in (7), $\mathbf{S}_m^l$ is a local part of $\mathbf{S}_m^g$.

Since all $\mathbf{G}_m$ are independent of each other, $\sum_{m=1}^{p}\min_{\mathbf{G}_m}\mathcal{L}(\mathbf{G}_m)$ leads to $\min_{\mathbf{G}}\sum_{m=1}^{p}\mathcal{L}(\mathbf{G}_m)$, and we further present a *group updating scheme*, which updates $\mathbf{G}$ by groups, that is, $\mathbf{G}_m$; then, (15) becomes the following one:

$$\max_{\mathbf{G}_m}\ \operatorname{tr}\Big(\Big(\mathbf{R}^\top\mathbf{B}_m + \alpha_1 r\mathbf{B}\mathbf{S}_m^g + \alpha_2 r\mathbf{B}_m\mathbf{S}_m^l + \mathbf{P}^\top\Big(\beta_1\mathbf{C}^g + \beta_2\mathbf{C}_m^l\Big)\mathbf{Y}_m\Big)\mathbf{G}_m^\top\Big)$$
$$\text{s.t.}\ \ \mathbf{G}_m\mathbf{G}_m^\top = n_m\mathbf{I},\ \ \mathbf{G}_m\mathbf{1} = \mathbf{0}. \tag{16}$$

It is worth noting that the optimal solution of $\sum_{m=1}^{p}\min_{\mathbf{G}_m}\mathcal{L}(\mathbf{G}_m)$ makes $\min_{\mathbf{G}}\sum_{m=1}^{p}\mathcal{L}(\mathbf{G}_m)$ reach its optimum; whereas the reverse is not necessarily satisfied. In other words, the solution of $\min_{\mathbf{G}}\sum_{m=1}^{p}\mathcal{L}(\mathbf{G}_m)$ is not necessarily the solution of $\sum_{m=1}^{p}\min_{\mathbf{G}_m}\mathcal{L}(\mathbf{G}_m)$. Therefore, this group updating scheme also considers the data distribution, embedding more information into the hash learning process.

To solve (16), we first define $\mathbf{Z}$ as follows:

$$\mathbf{Z} = \mathbf{R}^\top\mathbf{B}_m + \alpha_1 r\mathbf{B}\mathbf{S}_m^g + \alpha_2 r\mathbf{B}_m\mathbf{S}_m^l + \mathbf{P}^\top\Big(\beta_1\mathbf{C}^g + \beta_2\mathbf{C}_m^l\Big)\mathbf{Y}_m. \tag{17}$$

Further denote $\mathbf{J} = \mathbf{I} - (1/n_m)\mathbf{1}\mathbf{1}^\top$; in fact, it is the right-hand side of $\mathbf{J}$ that will be used in calculations. Then, we perform eigendecomposition of $\mathbf{Z}\mathbf{J}\mathbf{Z}^\top$ as follows:

$$\mathbf{Z}\mathbf{J}\mathbf{Z}^\top = \Big[\mathbf{V}\ \hat{\mathbf{V}}\Big]\begin{bmatrix}\Sigma & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\Big[\mathbf{V}\ \hat{\mathbf{V}}\Big] \tag{18}$$

where $\Sigma \in \Re^{r'\times r'}$ is the diagonal matrix of positive eigenvalues and $\mathbf{V} \in \Re^{r\times r'}$ is the corresponding eigenvectors. r' is the rank of $\mathbf{Z}\mathbf{J}\mathbf{Z}^\top$. $\hat{\mathbf{V}}$ is the matrix of remaining $r - r'$ eigenvectors with zero eigenvalue. Define $\mathbf{U} = \mathbf{J}\mathbf{Z}^\top\mathbf{V}\Sigma^{-1/2}$. We then perform the Gram–Schmidt process on $\hat{\mathbf{V}}$ and a random matrix to generate an orthogonal matrix $\bar{\mathbf{V}} \in \Re^{r\times(r-r')}$ and an orthogonal random matrix $\bar{\mathbf{U}} \in \Re^{n_m\times(r-r')}$, respectively. Thereafter, according to [14], the solution of (16) is

$$\mathbf{G}_m = \sqrt{n_m}\big[\mathbf{V}\quad \bar{\mathbf{V}}\big]\big[\mathbf{U}\quad \bar{\mathbf{U}}\big]. \tag{19}$$

If $r' = r$, $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ are empty. After all $\mathbf{G}_m(m \in \{1, 2, \ldots, p\})$ are updated, the entire $\mathbf{G} = [\mathbf{G}_1, \mathbf{G}_2, \ldots, \mathbf{G}_p]$ is obtained. It is worth noting that each group can be updated in parallel, which will further improve the updating efficiency.

***R**-Step:* Fixing all variables but $\mathbf{R}$, (13) becomes the following one:

$$\max_{\mathbf{R}}\operatorname{tr}\Big(\mathbf{R}^\top\mathbf{B}\mathbf{G}^\top\Big),\ \ \text{s.t.}\ \mathbf{R}\mathbf{R}^\top = \mathbf{I}. \tag{20}$$

The above problem is a classical orthogonal procrustes problem [18], and can be solved by performing singular value decomposition (SVD) on $\mathbf{B}\mathbf{G}^\top$ to obtain $\mathbf{A}\Omega\hat{\mathbf{A}}^\top$. Thereafter, the optimal $\mathbf{R}$ is

$$\mathbf{R} = \mathbf{A}\hat{\mathbf{A}}^\top. \tag{21}$$

***B**-Step:* Similar to the update progress of $\mathbf{G}$, $\mathbf{B}$ also can be updated group by group, that is, $\mathbf{B}_m$. When all variables but $\mathbf{B}$ are fixed, (13) is rewritten as

$$\max_{\mathbf{B}_m}\ \operatorname{tr}\Big(\Big(\mathbf{R}\mathbf{G}_m + \alpha_1 r\mathbf{G}\mathbf{S}_m^g + \alpha_2 r\mathbf{G}_m\mathbf{S}_m^l\Big)\mathbf{B}_m^\top\Big)$$
$$\text{s.t.}\ \ \mathbf{B}_m \in \{-1, 1\}^{r\times n_m}. \tag{22}$$

Algorithm 1 FCMH

```
Input: Training data $\mathbf{X}^{(k)}$ ($k \in \{1, 2, ..., o\}$), label matrix $\mathbf{Y}$,
    code length $r$, maximum iteration number $t$, parameters $\alpha_1$,
    $\alpha_2$, $\beta_1$, $\beta_2$, $\gamma$, $\xi$.
Output: Hash codes $\mathbf{B}$, hash function $H_k(\mathbf{x}^{(k)}_{query})$.
  % Step-1: Hash Codes Learning.
  1. Initialize $\mathbf{B}$, $\mathbf{G}$ randomly with a standard normal distribution;
  repeat
    2. Sequentially update $\mathbf{P}$, $\mathbf{G}$, $\mathbf{R}$, and $\mathbf{B}$ using (14), (19),
    (21), and (23), respectively.
  until convergent or maximum iterations
  % Step-2: Hash Functions Learning.
  3. Construct the mapping matrix $\mathbf{W}^{(k)}$ using (25);
  return Hash codes $\mathbf{B}$; hash function $H_k(\mathbf{x}^{(k)}_{query})$.
```

The optimal solution to the above problem is

$$\mathbf{B}_m = \operatorname{sign}\Big(\mathbf{R}\mathbf{G}_m + \alpha_1 r\mathbf{G}\mathbf{S}_m^g + \alpha_2 r\mathbf{G}_m\mathbf{S}_m^l\Big). \tag{23}$$

By generating binary codes for all groups, we have the entire $\mathbf{B} = [\mathbf{B}_1, \mathbf{B}_2, \ldots, \mathbf{B}_p]$.

C. Hash Functions Learning

After obtaining the hash codes, we then learn the hash functions for out-of-sample instances to map original data into the Hamming space. It can be solved by a binary classification problem with the supervision of the learned binary codes [44]. Many classification models can be applied in this problem. Typical examples include, but are not limited to, linear regression, support vector machine, and deep neural networks. Thereinto, linear regression is the simplest and most widely used model [45]. Specifically, given the learned hash codes, the mapping matrix for the kth modality can be obtained by the following objective function:

$$\min_{\mathbf{W}^{(k)}}\Big\|\mathbf{B} - \mathbf{W}^{(k)}\mathbf{X}^{(k)}\Big\|^2 + \xi\Big\|\mathbf{W}^{(k)}\Big\|^2 \tag{24}$$

where ξ is a penalty parameter to avoid overfitting. By setting the derivative of (24) with respect to $\mathbf{W}^{(k)}$ to 0, the optimal $\mathbf{W}^{(k)}$ is

$$\mathbf{W}^{(k)} = \mathbf{B}\mathbf{X}^{(k)\top}\Big(\mathbf{X}^{(k)}\mathbf{X}^{(k)\top} + \xi\mathbf{I}\Big)^{-1}. \tag{25}$$

Thereafter, the hash function for a new query of the kth modality is

$$H_k\Big(\mathbf{x}^{(k)}_{\text{query}}\Big) = \operatorname{sign}\Big(\mathbf{W}^{(k)}\mathbf{x}^{(k)}_{\text{query}}\Big). \tag{26}$$

To have an overall view, the entire training process of FCMH, including hash codes learning and hash functions learning, is summarized in Algorithm 1.

D. Convergence Proof

Based on the theory in [46], we give a theoretical convergence analysis of FCMH. First, the convergence of the hash codes optimization algorithm can be proved as follows. Denote $\mathcal{L}(\mathbf{P}, \mathbf{G}, \mathbf{R}, \mathbf{B})$ as the objective function in (12). As described in Section III-B4, all variables have a closed-form solution in their corresponding subproblems; therefore, the alternative updating rule will lead to $\mathcal{L}(\mathbf{P}^{t'+1}, \mathbf{G}^{t'+1}, \mathbf{R}^{t'+1}, \mathbf{B}^{t'+1}) \leq \mathcal{L}(\mathbf{P}^{t'+1}, \mathbf{G}^{t'+1}, \mathbf{R}^{t'+1}, \mathbf{B}^{t'}) \leq \mathcal{L}(\mathbf{P}^{t'+1}, \mathbf{G}^{t'+1}, \mathbf{R}^{t'}, \mathbf{B}^{t'}) \leq \mathcal{L}(\mathbf{P}^{t'+1}, \mathbf{G}^{t'}, \mathbf{R}^{t'}, \mathbf{B}^{t'}) \leq \mathcal{L}(\mathbf{P}^{t'}, \mathbf{G}^{t'}, \mathbf{R}^{t'}, \mathbf{B}^{t'})$, where t' is the iterative round. In other words, the objective value is monotonously decreasing in each alternative updating. Besides, the objective function in (12) is the summation of positive norms, which is lower bounded by 0. Therefore, after several iterations, the algorithm is able to converge to a stable solution. Second, for the hash functions learning step, (24) has a closed-form solution. Overall, the entire learning algorithm of FCMH is theoretically convergent.

E. Complexity Analysis

In this section, we analyze the space and time complexity of FCMH. The size of all variables and temporary variables, for example, $\mathbf{X}^{(k)}$, $\mathbf{Y}$, $\mathbf{S}^g$, $\mathbf{S}^l_m$, $\mathbf{C}^g$, $\mathbf{C}^l_m$, $\mathbf{D}^g$, $\mathbf{D}^l_m$, $\mathbf{B}$, $\mathbf{G}$, $\mathbf{P}$, $\mathbf{R}$, and $\mathbf{W}^{(k)}$, are linear to n or irrelevant to n, where n is the number of training instances. It is worth noting that $\mathbf{S}^g$ and $\mathbf{S}^l_m$ are calculated in an online mode by using the right-hand sides of (3) and (7), respectively. Hence, the use of $n \times n$ matrix can be avoided; thereafter, the entire space complexity of the training step is linear to n, that is, $O(n)$.

The time complexity includes $O(rc^2 + rc + c^2 + rcn)$ for updating $\mathbf{P}$, $O(r^3 + r^2 + r^2n + c^2n + rcn + rn)$ for updating $\mathbf{G}$, $O(r^3 + r^2n)$ for updating $\mathbf{R}$, $O(r^2n + rcn)$ for updating $\mathbf{B}$, and $O(d_k^2 + rd_k^2 + d_k^2n + rd_kn)$ for calculating $\mathbf{W}^{(k)}$, respectively, where r is the hash code length, c is the number of categories, and d_k is the feature dimensionality of kth modality. Therefore, the overall time complexity is $O(t(r^2 + c^2 + rc + r)n + o(d_k^2 + rd_k)n)$ for training, where t is the maximum iterations, and o is the number of modalities. Since r, c, d_k, t, $o << n$, the time complexity of FCMH is linear to the size of training set, that is, $O(n)$.

IV. Experiments

A. Experimental Settings

1) Datasets: To evaluate the performance of FCMH, we conducted experiments on three multilabel datasets, that is: 1) IAPR TC-12 [47]; 2) MIRFlickr-25K [48]; and 3) NUS-WIDE [49], which are widely used in the existing cross-modal hashing literature.

IAPR TC-12: It contains 20 000 image–text pairs, annotated by 255 labels. Each image and text is represented by a 512-D GIST feature vector and a 2912-D bag-of-words vector, respectively. We randomly selected 2000 pairs as the query set and the remaining as the retrieval and training sets.

MIRFlickr-25K: It is collected from Flickr, including 25 000 instances. Each instance belongs to at least one of 24 categories. The image is represented by a 512-D GIST feature vector, while the text is represented by a 1386-D bag-of-words vector. Similar to [30], those instances that have no labels are removed. Consequently, 20 015 instances are left. We randomly selected 2000 instances as the query set and the rest as the retrieval and training sets.

NUS-WIDE: It includes 269 468 real-world images of 81 categories. The image is represented by a 500-D SIFT feature vector. Each image is associated with a text caption, represented by a 1000-D binary tagging vector. Following [30], we only utilized 186 577 instances that belong to the top-10 most frequent labels. Finally, 2000 instances are taken as the query set and the rest as the retrieval and training sets.

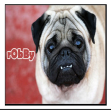
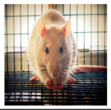
Modality	Query	Ground-truth neighbors (randomly selected)		
Image				
Text	dog, pet	dog	yellow, cute, duck	lomo, pet, picnik
Label	**animals**, **dog**	**animals**, **dog**, plant_life	**animals**, bird, indoor, people	**animals**, indoor

Fig. 2. Example of the query and corresponding ground-truth neighbors on MIRFlickr-25K. The common labels are marked in boldface.

For all three datasets, the ground-truth neighbors are defined as those image–text pairs sharing at least one common label with the query. Fig. 2 illustrates an example of the query and corresponding ground-truth neighbors in MIRFlickr-25K.

2) Evaluation Metrics: We conducted two cross-modal retrieval tasks, that is: 1) image-to-text ($I \rightarrow T$) and 2) text-to-image ($T \rightarrow I$). Thereinto, $I \rightarrow T$ means using images as the query to retrieve texts, and $T \rightarrow I$ means using texts as the query to retrieve images. In this article, several widely used evaluation protocols, including mean average precision (MAP), normalized discounted cumulative gain (NDCG), precision–recall curve, top-N precision curve, and training time, are utilized to measure the retrieval performance of different methods.

Given a query, the average precision (AP) is defined as

$$\mathrm{AP} = \frac{1}{n_p} \sum_{v=1}^{n_r} \mathrm{P}(v)\theta(v) \tag{27}$$

where n_p is the number of ground-truth neighbors in the retrieval set, n_r is the size of the retrieval set, and $\mathrm{P}(v)$ indicates the precision of top-v retrieved instances. $\theta(v)$ is an indicator function, which equals 1 if the vth retrieved instance is relevant to the query, and 0 otherwise. Then, MAP is defined as

$$\mathrm{MAP} = \frac{1}{n_q} \sum_{i=1}^{n_q} \mathrm{AP}(i) \tag{28}$$

where n_q is the size of the query set and $\mathrm{AP}(i)$ is the AP of the ith instance in the query set. The larger the MAP, the better the retrieval accuracy.

For multilabel data, discounted cumulative gain (DCG) is used to evaluate the ranking performance in retrieval tasks, which is defined as

$$\mathrm{DCG} = \sum_{i=1}^{n_r} \frac{2^{\mathrm{rel}_i} - 1}{\log_2(1+i)}. \tag{29}$$

rel_i should indicate the semantic relevance of the retrieval list at position i. For this purpose, we define it as follows:

$$\mathrm{rel}_i = \mathbf{y}_{\mathrm{query}}^{\top} \mathbf{C}^g \mathbf{y}_i \tag{30}$$

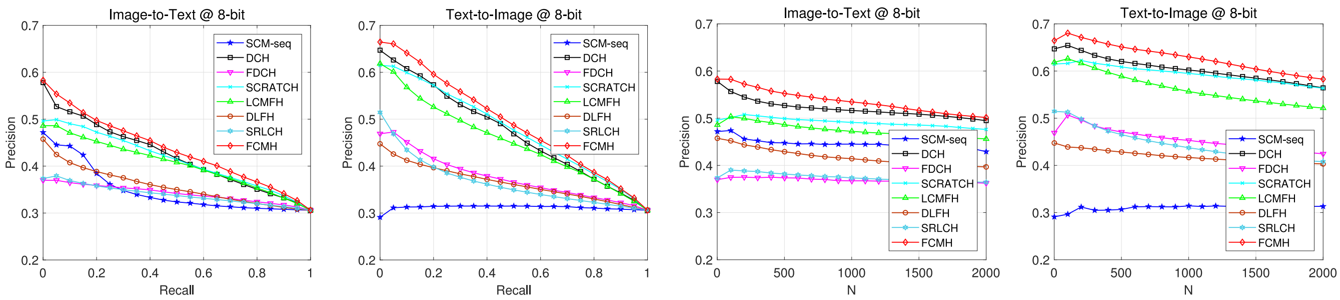


Fig. 3. Precision–recall and top-N precision curves of all methods on IAPR TC-12.

TABLE I
MAP RESULTS OF FCMH AND BASELINES ON IAPR TC-12

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \to T$	SCM-seq	0.3446	0.3983	0.4256	0.4344	0.3875
	DCH	0.4445	0.4663	0.4810	0.5010	0.5154
	FDCH	0.3527	0.3587	0.3750	0.4168	0.4225
	SCRATCH	0.4490	0.4606	0.4765	0.4896	0.4906
	LCMFH	0.4196	0.4273	0.4472	0.4584	0.4651
	DLFH	0.3791	0.4064	0.4586	0.4843	0.5047
	SRLCH	0.3484	0.3847	0.3942	0.4269	0.4388
	FCMH	**0.4624**	**0.4878**	**0.5074**	**0.5245**	**0.5348**
$T \to I$	SCM-seq	0.3097	0.3515	0.3990	0.4028	0.3607
	DCH	0.4965	0.5383	0.5730	0.6017	0.6295
	FDCH	0.3956	0.4275	0.4750	0.5314	0.5675
	SCRATCH	0.5076	0.5482	0.5794	0.6105	0.6240
	LCMFH	0.4649	0.4993	0.5261	0.5546	0.5662
	DLFH	0.3928	0.4496	0.5213	0.5951	0.6312
	SRLCH	0.3745	0.4274	0.4533	0.5000	0.5293
	FCMH	**0.5347**	**0.5879**	**0.6253**	**0.6589**	**0.6764**

TABLE II
NDCG RESULTS OF FCMH AND BASELINES ON IAPR TC-12

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \to T$	SCM-seq	0.9050	0.9155	0.9200	0.9221	0.9144
	DCH	0.9190	0.9254	0.9258	0.9296	0.9309
	FDCH	0.9022	0.9014	0.9038	0.9129	0.9129
	SCRATCH	0.9142	0.9196	0.9237	0.9263	0.9278
	LCMFH	0.9154	0.9180	0.9223	0.9258	0.9267
	DLFH	0.9031	0.9119	0.9171	0.9232	0.9255
	SRLCH	0.9016	0.9076	0.9116	0.9181	0.9206
	FCMH	**0.9198**	**0.9254**	**0.9288**	**0.9335**	**0.9367**
$T \to I$	SCM-seq	0.8930	0.9031	0.9140	0.9148	0.9075
	DCH	0.9263	0.9345	0.9411	0.9471	0.9494
	FDCH	0.9134	0.9176	0.9263	0.9359	0.9396
	SCRATCH	0.9244	0.9335	0.9404	0.9458	0.9489
	LCMFH	0.9276	0.9348	0.9402	0.9456	0.9476
	DLFH	0.9054	0.9189	0.9273	0.9373	0.9420
	SRLCH	0.9097	0.9175	0.9253	0.9324	0.9366
	FCMH	**0.9298**	**0.9373**	**0.9420**	**0.9474**	**0.9512**

where $\mathbf{y}_{\text{query}}$ and $\mathbf{y}_i$ are labels of the given query and the retrieved point with rank i, respectively. $\mathbf{C}^g$ is the global class correlation matrix defined in Section III-B1. The larger the rel_i, the more similar the ith retrieved point with the query. Thereafter, NDCG is defined as

$$\text{NDCG} = \text{DCG}/\text{IDCG} \tag{31}$$

where IDCG is the ideal DCG for a query, which can be calculated by the ideal relevance ranking. The larger the NDCG, the higher relevant points are ranked at the top of the retrieval list.

3) Baselines and Implementation Details: We compared FCMH with seven state-of-the-art supervised cross-modal hashing methods, that is: 1) SCM-seq [12]; 2) DCH [8]; 3) FDCH [9]; 4) SCRATCH [31]; 5) LCMFH [10]; 6) DLFH [30]; and 7) SRLCH [11]. For FDCH, LCMFH, and SRLCH, we implemented them by ourselves with the suggested parameters in their papers. All other baselines are carefully implemented based on the source codes provided by their authors. For the proposed FCMH, we first conducted a parameter experiment on MIRFlickr-25K. Based on the results shown in Section IV-B4, the values of parameters that yield the best performance are selected. For simplicity, we set the same parameter values for all benchmark datasets, that is, $\alpha_1 = 100$, $\alpha_2 = 10$, $\beta_1 = 10$, and $\beta_2 = 10$. In addition, the regularization parameters γ and ξ are empirically set to 0.1 and 1, respectively. The maximum iterative number t is set to 5. To generate groups of data, we adopted a multiview K-means clustering algorithm [50]. Its time complexity is $O(dn)$, where d and n are the dimensionality and number of samples, respectively. In order to further reduce the time cost of clustering, we reduced the dimensionality of data into 200 via principal component analysis [51]; then, training samples are empirically separated into $\min(25, c)$ clusters, where c is the number of categories. Notably, the reduced data are only used in the clustering process. Other clustering methods can also be utilized. When datasets are big enough, differences among clustering methods are slight. All our experiments are performed on a workstation with Intel Xeon E5-2650 CPU @2.20 GHz, 128-GB RAM.

B. Results and Discussion

1) Results on IAPR TC-12: The MAP and NDCG results of FCMH and all baselines on IAPR TC-12 are reported in Tables I and II, respectively. They include the image-to-text and text-to-image tasks. The code length varies from 8 to 128. The best results are shown in boldface. Besides, the precision-recall and top-N precision curves with the case of 8 bits are plotted in Fig. 3. From these results, we have the following observations.

1) As shown in Table I, FCMH outperforms all baselines in all cases on this dataset. Compared with several recent methods, for example, SCRATCH, LCMFH, and DLFH, the MAP results of FCMH are still competitive. In detail, FCMH obtains about 2% and 4% performance gains over the best baseline on image-to-text and text-to-image tasks, respectively, demonstrating the effectiveness of FCMH in exploiting supervised information.

TABLE III
MAP RESULTS OF FCMH AND BASELINES ON MIRFLICKR-25K

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.6413	0.6557	0.6670	0.6730	0.6356
	DCH	0.6913	0.6971	0.7178	0.7190	0.7412
	FDCH	0.6405	0.6883	0.6612	0.6785	0.7143
	SCRATCH	0.7130	0.7214	0.7182	0.7292	0.7332
	LCMFH	0.6725	0.6854	0.6941	0.6930	0.7015
	DLFH	0.6840	0.7116	0.7243	0.7331	0.7348
	SRLCH	0.6431	0.6408	0.6651	0.6649	0.6800
	FCMH	**0.7242**	**0.7470**	**0.7522**	**0.7564**	**0.7580**
$T \rightarrow I$	SCM-seq	0.6294	0.6365	0.6536	0.6638	0.6216
	DCH	0.7462	0.7601	0.7748	0.7875	0.8074
	FDCH	0.6900	0.7572	0.7506	0.7600	0.7989
	SCRATCH	0.7577	0.7879	0.7749	0.7936	0.8006
	LCMFH	0.7271	0.7496	0.7643	0.7625	0.7819
	DLFH	0.7629	0.8009	0.8206	0.8358	0.8420
	SRLCH	0.6717	0.6825	0.7098	0.7061	0.7302
	FCMH	**0.8075**	**0.8294**	**0.8423**	**0.8487**	**0.8515**

TABLE IV
NDCG RESULTS OF FCMH AND BASELINES ON MIRFLICKR-25K

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.9243	0.9272	0.9288	0.9299	0.9238
	DCH	0.9310	0.9337	0.9380	0.9377	0.9441
	FDCH	0.9172	0.9292	0.9231	0.9259	0.9349
	SCRATCH	0.9308	0.9313	0.9327	0.9351	0.9354
	LCMFH	0.9289	0.9311	0.9322	0.9327	0.9338
	DLFH	0.9197	0.9230	0.9265	0.9269	0.9262
	SRLCH	0.9239	0.9195	0.9255	0.9263	0.9289
	FCMH	**0.9357**	**0.9392**	**0.9411**	**0.9423**	**0.9450**
$T \rightarrow I$	SCM-seq	0.9226	0.9222	0.9261	0.9279	0.9209
	DCH	0.9394	0.9434	0.9459	0.9472	0.9526
	FDCH	0.9283	0.9405	0.9368	0.9389	0.9461
	SCRATCH	0.9366	0.9419	0.9403	0.9433	0.9451
	LCMFH	0.9386	0.9424	0.9431	0.9435	0.9461
	DLFH	0.9326	0.9386	0.9418	0.9425	0.9436
	SRLCH	0.9302	0.9276	0.9327	0.9329	0.9360
	FCMH	**0.9464**	**0.9497**	**0.9525**	**0.9533**	**0.9545**

TABLE V
MAP RESULTS OF FCMH AND BASELINES ON NUS-WIDE

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.4981	0.4599	0.5492	0.5410	0.5419
	DCH	0.6090	0.5843	0.6268	0.5533	0.6016
	FDCH	0.5901	0.6065	0.5957	0.6236	0.6322
	SCRATCH	0.6190	0.6381	0.6392	0.6484	0.6524
	LCMFH	0.5805	0.5878	0.6299	0.6387	0.6441
	DLFH	0.5892	0.6321	0.6597	0.6694	0.6787
	SRLCH	0.5988	0.6119	0.6362	0.6545	0.6618
	FCMH	**0.6573**	**0.6655**	**0.6763**	**0.6820**	**0.6818**
$T \rightarrow I$	SCM-seq	0.4784	0.4525	0.5219	0.5220	0.5268
	DCH	0.7132	0.7058	0.7487	0.6598	0.7255
	FDCH	0.7171	0.7607	0.7773	0.7833	0.7926
	SCRATCH	0.7255	0.7437	0.7612	0.7790	0.7830
	LCMFH	0.6673	0.6708	0.7222	0.7357	0.7450
	DLFH	0.7190	0.7584	0.7929	0.8017	0.8034
	SRLCH	0.7183	0.7373	0.7640	0.7763	0.7941
	FCMH	**0.7557**	**0.7767**	**0.8022**	**0.8069**	**0.8170**

TABLE VI
NDCG RESULTS OF FCMH AND BASELINES ON NUS-WIDE

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.9266	0.9220	0.9329	0.9321	0.9323
	DCH	0.9396	0.9353	0.9411	0.9300	0.9380
	FDCH	0.9334	0.9347	0.9355	0.9372	0.9395
	SCRATCH	0.9398	0.9426	0.9422	0.9443	0.9451
	LCMFH	0.9353	0.9347	0.9400	0.9423	0.9433
	DLFH	0.9335	0.9397	0.9439	0.9449	0.9462
	SRLCH	0.9368	0.9390	0.9418	0.9445	0.9454
	FCMH	**0.9436**	**0.9458**	**0.9477**	**0.9486**	**0.9482**
$T \rightarrow I$	SCM-seq	0.9280	0.9252	0.9355	0.9356	0.9364
	DCH	0.9567	0.9571	0.9619	0.9505	0.9597
	FDCH	0.9556	0.9609	0.9642	0.9631	0.9662
	SCRATCH	0.9553	0.9588	0.9604	0.9623	0.9626
	LCMFH	0.9493	0.9491	0.9561	0.9586	0.9600
	DLFH	0.9526	0.9581	0.9649	0.9654	0.9666
	SRLCH	0.9565	0.9589	0.9607	0.9623	0.9648
	FCMH	**0.9615**	**0.9644**	**0.9672**	**0.9683**	**0.9682**

2) Concerning the NDCG results, FCMH achieves the best results with all code lengths, indicating that FCMH is able to rank those semantic relevant instances at the top of the retrieval list. One possible reason is that FCMH explores fine-grained semantics, that is, class correlation and local similarity, making the retrieval results much more fine-grained than baselines.

3) With the code length increasing, most methods perform better, confirming the fact that longer bits can carry more supervised information. More importantly, FCMH obtains better performance with short code length than most baselines with long code length. This phenomenon indicates that FCMH has a strong information embedding capability to generate discriminative hash codes even with short code length.

4) From Fig. 3, we can observe that FCMH outperforms all baselines, which is consistent with the MAP results, further demonstrating the effectiveness of FCMH.

2) Results on MIRFlickr-25K: The MAP results of all methods on MIRFlickr-25K are summarized in Table III, followed by the NDCG results in Table IV. In addition, we further plotted the precision-recall and top-N precision curves in Fig. 4. From these results, we have similar observations to those on IAPR TC-12.

1) FCMH achieves the best MAP and NDCG results in all cases on this dataset. The success of FCMH is due to its exploitation of fine-grained supervised information through pairwise similarity preserving and correlated label reconstructing, and due to the utilization of both global and local similarities of data.

2) SCRATCH, LCMFH, and DLFH are the most recent three baselines; however, FCMH still outperforms them by a large gap, demonstrating its effectiveness.

3) FCMH obtains larger improvements over the best baseline on the text-to-image task than on the image-to-text task. One possible reason is that FCMH can fully utilize the local similarity hidden in rich textual features.

4) Generally speaking, the performance of methods on the text-to-image task is better than that on the image-to-text task. The main reason is that the textual features carry more information than the handcrafted features of image modality. By using deep image features, this difference will disappear, which can be observed in Table IX.

3) Results on NUS-WIDE: We summarized the MAP and NDCG results of FCMH and all baselines on NUS-WIDE in Tables V and VI, respectively. Furthermore, we plotted the precision–recall and top-*N* precision curves in Fig. 5. From these results, we have similar observations to those on IAPR TC-12 and MIRFlickr-25K.

1) From all the results, we can see that FCMH consistently outperforms all baselines, demonstrating FCMH is able to generate more discriminative hash codes through the

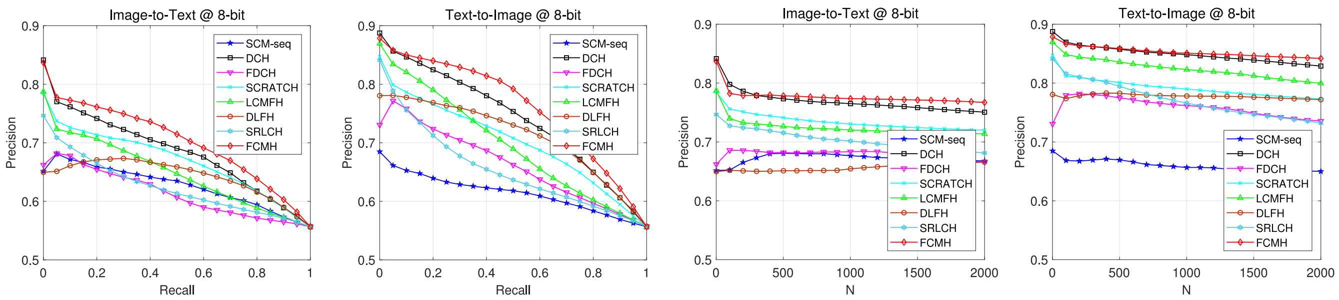


Fig. 4. Precision–recall and top-N precision curves of all methods on MIRFlickr-25K.

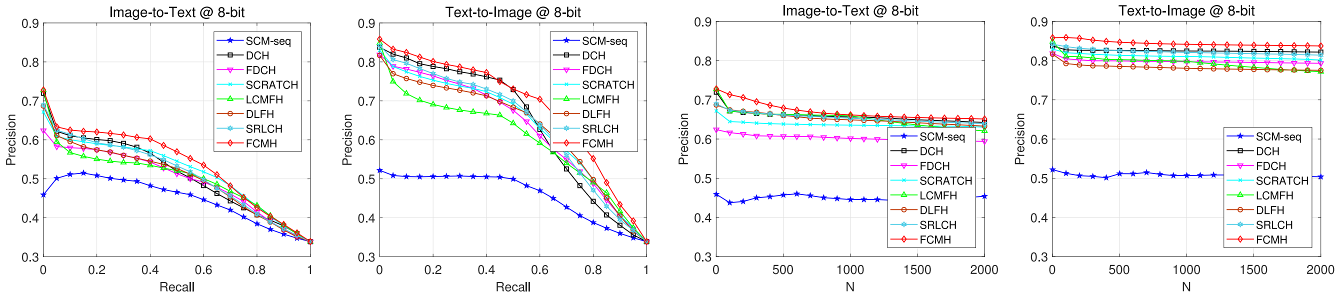
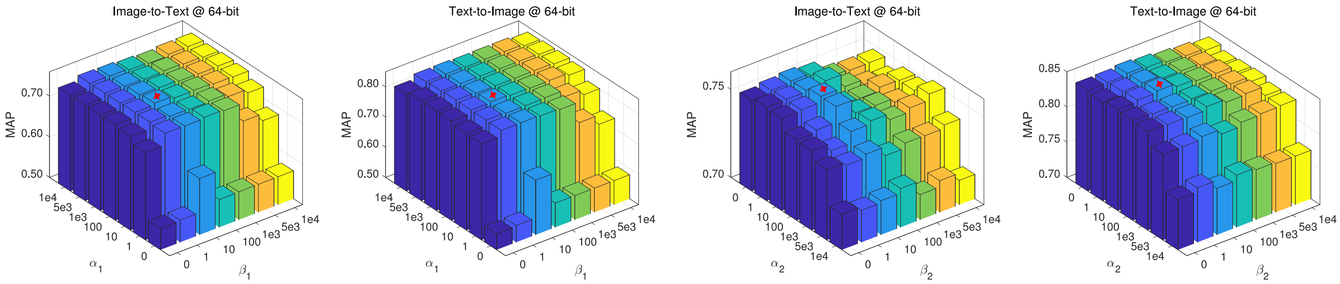


Fig. 5. Precision–recall and top-N precision curves of all methods on NUS-WIDE.



(a)

(b)

Fig. 6. Sensitivity analysis of parameters α_1, α_2, β_1, and β_2 on MIRFlickr-25K. (a) $\alpha_2 = 0$ and $\beta_2 = 0$. (b) $\alpha_1 = 100$ and $\beta_1 = 10$.

proposed global and local similarity embedding framework. Especially, the NDCG results of FCMH are higher than all baselines, indicating that the retrieval list of FCMH is more fine grained, that is, more relevant samples are preferentially returned in the retrieval list.

2) In considering the MAP results, FCMH obtains more performance gains with short code length on both image-to-text and text-to-image tasks. Specifically, it achieves about 3% improvements over the best baseline in the cases of 8 and 16 bits, demonstrating the learning capability of FCMH.

3) In all baselines, SCM-seq and DLFH exploit the pairwise similarity, while others leverage the label matrix. None of them explores the class correlation and local structure of data. In contrast, FCMH explores fine-grained supervised information of data, including globally and locally extracted pairwise similarities and class correlations; therefore, it generates more discriminative hash codes and obtains better retrieval performance.

4) Parameter Sensitivity Analysis: We also conducted experiments on MIRFlickr-25K to analyze the performance of FCMH with the parameters varying, including α_1, α_2 and β_1, β_2. Notably, γ and ξ are for regularization, which can be set empirically; thus, we did not perform analyses on them. The MAP results of FCMH on both image-to-text and text-to-image tasks are plotted in Fig. 6. The code length is 64 bits. The best results are marked with red stars. First, we plotted the parameters analysis of α_1 and β_1 while fixing $\alpha_2 = 0$ and $\beta_2 = 0$ in Fig. 6(a). We can observe that the parameters indeed have some influence on the performance of FCMH, and FCMH obtains the best results when $\alpha_1 = 100$ and $\beta_1 = 10$. Second, we plotted the MAP results by varying α_2 and β_2 from 0 to $1e4$ with $\alpha_1 = 100$ and $\beta_1 = 10$ in Fig. 6(b). From this figure, we can see that the parameters also affect the performance of FCMH, and the best performance is achieved at $\alpha_2 = 10$ and $\beta_2 = 10$. Overall, the best performance is achieved at $\alpha_1 = 100$, $\beta_1 = 10$, $\alpha_2 = 10$, $\beta_2 = 10$.

To some extent, the parameters also reflect the role of different items in the objective function, that is, (12). For example, in Fig. 6(a), when α_1 varying from 0 to $1e4$, its performance has large improvements, indicating the effectiveness of the global pairwise similarity preserving. Similarly, there is a large

TABLE VII
TRAINING TIME (SECOND) OF FCMH AND BASELINES ON THREE DATASETS

Method	IAPR TC-12					MIRFlickr-25K					NUS-WIDE				
	8-bit	16-bit	32-bit	64-bit	128-bit	8-bit	16-bit	32-bit	64-bit	128-bit	8-bit	16-bit	32-bit	64-bit	128-bit
SCM-seq	14.04	22.73	43.14	80.01	158.42	9.72	16.31	31.52	59.55	58.41	7.19	8.36	18.43	32.38	62.57
DCH	13.94	16.93	25.75	52.66	136.22	4.48	5.69	8.47	17.04	66.43	28.82	34.29	84.15	283.59	1217.60
FDCH	21.26	20.24	21.13	20.16	21.41	8.19	7.87	7.59	7.83	8.07	44.42	43.04	46.52	47.56	50.41
SCRATCH	3.49	3.60	3.80	4.64	6.04	2.20	2.25	2.40	3.15	4.54	19.51	19.21	23.07	28.88	42.87
LCMFH	16.54	15.20	15.29	15.65	17.23	4.41	4.39	4.57	5.06	5.91	23.39	24.68	25.84	29.33	36.64
DLFH	3.53	4.73	6.71	20.33	73.60	1.27	2.54	6.31	38.57	114.57	8.12	16.41	92.81	326.63	1260.36
SRLCH	9.06	8.90	8.92	9.20	9.63	7.97	7.85	8.08	8.51	8.77	67.02	68.60	68.83	73.04	72.15
FCMH	**3.05**	**3.25**	**3.35**	**4.19**	**5.58**	**1.07**	**1.02**	**1.18**	**1.53**	**2.40**	**5.27**	**5.87**	**6.59**	**9.10**	**14.13**

performance gain with β_1 ranging from 0 to $1e4$, confirming the effectiveness of the global class correlation. In addition, the performance of FCMH with $\beta_1 = 0$ is much higher than those with $\alpha_1 = 0$, revealing that the pairwise similarity preserving plays a larger role than the correlated label reconstructing during optimization, perhaps caused by the orthogonal constraint in (12). From Fig. 6(b), we can find that with α_2 and β_2 varying from 1 to 100, the performance has some improvements, demonstrating the effectiveness of the local structure. However, when giving α_2 and β_2 an extremely big value, the performance dramatically degrades, which indicates that there is a balance between the local structure and global structure, and too much local structure will adversely affect retrieval performance. It is worth noting that all parametric experiments are conducted with the group updating scheme; thus, the local distribution is also considered, even when $\alpha_2 = 0$ and $\beta_2 = 0$.

5) Time Cost Analysis: In Section III-E, we show the time complexity of FCMH is linear to the size of the training set, which is scalable to large-scale datasets. To further demonstrate its scalability, we conducted experiments and recoded the training time of all methods on IAPR TC-12, MIRFlickr-25K, and NUS-WIDE. As mentioned previously, in FCMH, the group updating scheme can be performed in parallel, which will further improve the optimization efficiency. For a fair comparison, we recorded the time of FCMH with no parallel strategy. Besides, the reported time of FCMH does not include clustering because the group partitioning is independent of the hash learning process, and the time complexity of different clustering algorithms is significantly different. The code length varies from 8 to 128 bits. The results are summarized in Table VII, and the best results are marked in boldface. As shown in this table, although all methods claim their complexity is $O(n)$, there are obvious differences. The training time of several methods, for example, SCM, DCH, and DLFH, increases significantly with the code length increasing; whereas, the increase of FCMH is very slight. Among all methods, FCMH is the fastest one, confirming the fact that FCMH is much more efficient and scalable to large-scale datasets.

6) Convergence Analysis: In Section III-D, we show the learning process of FCMH is provably convergent. To have a deep insight, we further conducted experiments to demonstrate the convergence of the hash codes optimization algorithm. In detail, we plotted the normalized objective values with respect to the iterations on MIRFlickr-25K and NUS-WIDE in Fig. 7. From this figure, we can find that the optimization algorithm always converges within five iterations, demonstrating its convergence and efficiency.

TABLE VIII
ABLATION RESULTS OF FCMH ON MIRFLICKR-25K

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	FCMH-lc	0.7150	0.7269	0.7310	0.7416	0.7436
	FCMH-l	0.7182	0.7326	0.7338	0.7449	0.7463
	FCMH	**0.7242**	**0.7470**	**0.7522**	**0.7564**	**0.7580**
$T \rightarrow I$	FCMH-lc	0.7946	0.8150	0.8261	0.8378	0.8456
	FCMH-l	0.7950	0.8172	0.8318	0.8409	0.8489
	FCMH	**0.8075**	**0.8294**	**0.8423**	**0.8487**	**0.8515**

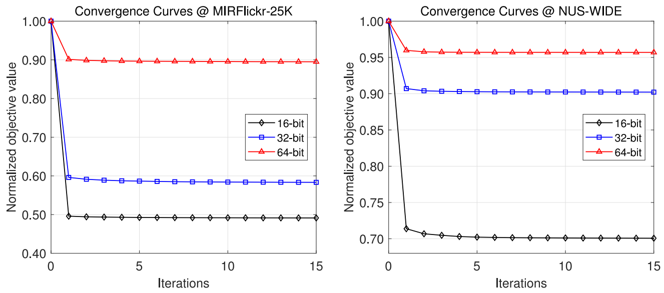


Fig. 7. Convergence analysis of the alternative optimization algorithm on MIRFlickr-25K and NUS-WIDE.

7) Ablation Study: To gain a deep insight into FCMH, we further conducted ablation experiments on MIRFlickr-25K, including two variations of FCMH, that is: 1) FCMH-lc and 2) FCMH-l. Thereinto, FCMH-lc is a base model that combines the pairwise similarity preserving objective and the label reconstructing objective and discretely solves the optimization problem without relaxation. Compared with FCMH, FCMH-lc drops out the local similarity and global class correlation. In contrast, FCMH-l discards the local similarity and takes the global class correlation into consideration. The MAP results of FCMH and its variations are reported in Table VIII. From this table, we can see the following.

1) FCMH-l has some performance improvements over FCMH-lc. This phenomenon indicates that the global class correlation is useful but not significant.
2) FCMH outperforms FCMH-lc and FCMH-l greatly, verifying the importance of local similarity, that is, local pairwise similarity and local class correlation.
3) FCMH-lc, as the worst variation, is still superior to all baselines in Table III, demonstrating the effectiveness of the base model.

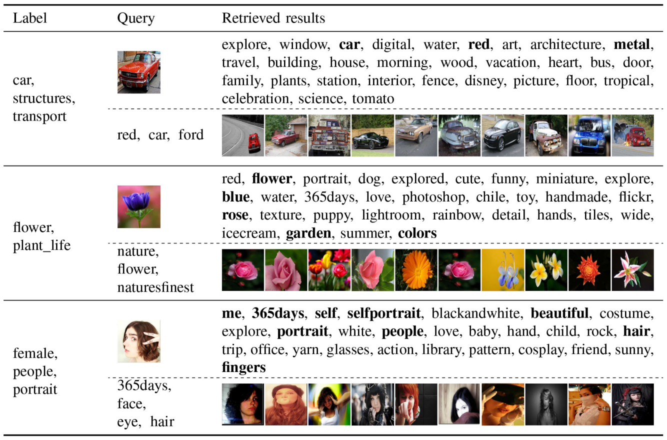

Label
Query
Retrieved results
car, structures, transport
explore, window, **car**, digital, water, **red**, art, architecture, **metal**, travel, building, house, morning, wood, vacation, heart, bus, door, family, plants, station, interior, fence, disney, picture, floor, tropical, celebration, science, tomato
red, car, ford
flower, plant_life
red, **flower**, portrait, dog, explored, cute, funny, miniature, explore, **blue**, water, 365days, love, photoshop, chile, toy, handmade, flickr, **rose**, texture, puppy, lightroom, rainbow, detail, hands, tiles, wide, icecream, **garden**, summer, **colors**
nature, flower, naturesfinest
female, people, portrait
me, **365days**, **self**, **selfportrait**, blackandwhite, **beautiful**, costume, explore, **portrait**, white, **people**, love, baby, hand, child, rock, **hair**, trip, office, yarn, glasses, action, library, pattern, cosplay, friend, sunny, **fingers**
365days, face, eye, hair

Fig. 8. Visualization results of FCMH on MIRFlickr-25K. Tags in boldface are manually marked as relevant.

8) Visualized Results: To have some visualized observations, we further showed the visualization results of FCMH on MIRFlickr-25K in Fig. 8. The hash code length is 64 bits. In Fig. 8, the middle consists of the queries, including three image-text pairs; the left consists of their ground-truth labels; the right corresponding retrieved results on image-to-text and text-to-image tasks, respectively. For each image query, all tags that appear in the top-10 retrieved texts are sorted according to their frequency of occurrence. We manually marked the relevant tags in boldface. For each text query, the top-10 retrieved images are presented, which are sorted based on the Hamming distance between hash codes. From these visualization results, we can observe that FCMH is able to retrieve samples that are semantically related to the query on both image-to-text and text-to-image tasks. Moreover, FCMH can preferentially return those visually similar samples from the semantically relevant samples, making the retrieval results more fine-grained.

9) Comparison With Deep Hashing: To further demonstrate the effectiveness of FCMH, we conducted experiments and compared it with five state-of-the-art deep cross-modal hashing methods, that is: 1) DCMH [52]; 2) SSAH [53]; 3) EGDH [54]; 4) MLCAH [55]; and 5) DADH [56], on MIRFlickr-25K. For a fair comparison, we utilized the deep image features extracted from a CNN-F deep network [57] pretrained on the ImageNet dataset [58]. Thereafter, FCMH is trained on 4096-D CNN features of image modality and 1386-D bag-of-words features of text modality, called FCMH$_{cnn}$. Following [52], we also randomly selected 2000 instances as the query set and the rest as the retrieval set. 10 000 instances randomly selected from the retrieval set are used as the training set. Other implementation details are as same as FCMH, for example, adopting linear regression in the hash functions learning step. The MAP results of FCMH and five baselines are summarized in Table IX. For all baselines, their results are that reported in the original papers. As shown in this table, FCMH obtains the best results in all cases. It is worth noting that FCMH$_{cnn}$ is not an end-to-end deep model, it still achieves very competitive results. One of the possible reasons is that although deep neural networks have strong nonlinear representation ability, they are usually not capable of optimizing complex objective functions, limiting their performance. On the contrary, due to the well-designed objective function and optimization algorithm, FCMH$_{cnn}$ achieves superior performance, demonstrating the effectiveness of the loss function and optimization algorithm.

TABLE IX
MAP RESULTS OF FCMH WITH CNN FEATURES AND DEEP BASELINES ON MIRFLICKR-25K

Task	Method	16-bit	32-bit	64-bit
$I \to T$	DCMH [52]	0.7410	0.7465	0.7485
	SSAH [53]	0.7820	0.7900	0.8000
	EGDH [54]	0.7569	0.7729	0.7959
	MLCAH [55]	0.7960	0.8080	0.8150
	DADH [56]	0.8020	0.8072	0.8179
	FCMH$_{cnn}$	**0.8200**	**0.8322**	**0.8412**
$T \to I$	DCMH [52]	0.7827	0.7900	0.7932
	SSAH [53]	0.7910	0.7950	0.8030
	EGDH [54]	0.7787	0.7939	0.7985
	MLCAH [55]	0.7940	0.8050	0.8050
	DADH [56]	0.7920	0.7959	0.8064
	FCMH$_{cnn}$	**0.8120**	**0.8235**	**0.8318**

V. CONCLUSION

In this article, we presented FCMH, focusing on how to effectively generate fine-grained retrieval results, how to discretely solve the binary constraint, and how to efficiently learn hash codes. In specific, it fully explores the supervised information through pairwise similarity preserving and correlated label reconstructing. In addition, it takes both global and local similarities of data into consideration through global and local similarity embedding. In light of this, its space and time complexity is much reduced, that is, linear to the size of the training set, which is efficient and scalable to large-scale datasets. In addition, it can embed more fine-grained supervised information into the to-be-learned hash codes, leading to more fine-grained retrieval results. Moreover, it discretely solves the binary optimization problem by a well-designed group updating scheme, reducing the quantization error. Extensive experimental results on three datasets demonstrated the effectiveness and efficiency of FCMH; especially, it outperforms some state-of-the-art shallow and deep cross-modal hashing methods.

ACKNOWLEDGMENT

The authors would like to thank the Editor, the Associate Editor, and the anonymous reviewers for their constructive comments and suggestions.

REFERENCES

[1] F. Shen, C. Shen, Q. Shi, A. V. D. Hengel, Z. Tang, and H. T. Shen, "Hashing on nonlinear manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1839–1851, Jun. 2015.

[2] Y. Liu *et al.*, "Deep self-taught hashing for image retrieval," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2229–2241, Jun. 2019.

[3] X. Nie, W. Jing, C. Cui, C. J. Zhang, L. Zhu, and Y. Yin, "Joint multi-view hashing for large-scale near-duplicate video retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 1951–1965, Oct. 2020.

[4] X. Xu, H. Lu, J. Song, Y. Yang, H. T. Shen, and X. Li, "Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2400–2413, Jun. 2020.

[5] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3864–3872.

[6] P. F. Zhang, C. X. Li, M. Y. Liu, L. Nie, and X. S. Xu, "Semi-relaxation supervised hashing for cross-modal retrieval," in *Proc. ACM Multimedia Conf.*, 2017, pp. 1762–1770.

[7] D. Mandal, K. N. Chaudhury, and S. Biswas, "Generalized semantic preserving hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 102–112, Jan. 2019.

[8] 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 Process.*, vol. 26, no. 5, pp. 2494–2507, May 2017.

[9] X. Liu, X. Nie, W. Zeng, C. Cui, L. Zhu, and Y. Yin, "Fast discrete cross-modal hashing with regressing from semantic labels," in *Proc. ACM Multimedia Conf.*, 2018, pp. 1662–1669.

[10] D. Wang, X. Gao, X. Wang, and L. He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2466–2479, Oct. 2019.

[11] H. T. Shen *et al.*, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 29, 2020, doi: 10.1109/TKDE.2020.2970050.

[12] D. Zhang and W. J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.

[13] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 37–45.

[14] W. Liu, C. Mu, S. Kumar, and S. F. Chang, "Discrete graph hashing," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.

[15] X. Yang, X. Gao, B. Song, and B. Han, "Hierarchical deep embedding for aurora image retrieval," *IEEE Trans. Cybern.*, early access, Jan. 10, 2020, doi: 10.1109/TCYB.2019.2959261.

[16] S. He *et al.*, "Bidirectional discrete matrix factorization hashing for image search," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 4157–4168, Sep. 2020.

[17] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Flexible multi-modal hashing for scalable multimedia retrieval," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 1–20, 2020.

[18] 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.

[19] J. Tang, Z. Li, M. Wang, and R. Zhao, "Neighborhood discriminant hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2827–2840, Sep. 2015.

[20] R. Ji, H. Liu, L. Cao, D. Liu, Y. Wu, and F. Huang, "Toward optimal manifold hashing via discrete locally linear embedding," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5411–5420, Nov. 2017.

[21] Y. Huang and Z. Lin, "Binary multidimensional scaling for hashing," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 406–418, Jan. 2018.

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

[23] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.

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

[25] X. Luo, P. F. Zhang, Z. Huang, L. Nie, and X.-S. Xu, "Discrete hashing with multiple supervision," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2962–2975, Jun. 2019.

[26] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 2075–2082.

[27] H. Liu, R. Ji, Y. Wu, F. Huang, and B. Zhang, "Cross-modality binary code learning via fusion similarity hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7380–7388.

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

[29] M. Hu, Y. Yang, F. Shen, N. Xie, R. Hong, and H. T. Shen, "Collective reconstructive embeddings for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2770–2784, Jun. 2019.

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

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

[32] J. Lu, V. E. Liong, and J. Zhou, "Deep hashing for scalable image search," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2352–2367, May 2017.

[33] Z. D. Chen, W. J. Yu, C. X. Li, L. Nie, and X. S. Xu, "Dual deep neural networks cross-modal hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 274–281.

[34] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Deep collaborative multi-view hashing for large-scale image search," *IEEE Trans. Image Process.*, vol. 29, pp. 4643–4655, 2020.

[35] X. Zhou *et al.*, "Graph convolutional network hashing," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1460–1472, Apr. 2020.

[36] Y. Xing, G. Yu, C. Domeniconi, J. Wang, and Z. Zhang, "Multi-label co-training," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2882–2888.

[37] S. J. Huang and Z. H. Zhou, "Multi-label learning by exploiting label correlations locally," in *Proc. AAAI Conf. Artif. Intell.*, 2012, pp. 949–955.

[38] Y. Zhu, J. T. Kwok, and Z. H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1081–1094, Jun. 2018.

[39] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[40] A. Gordo, F. Perronnin, Y. Gong, and S. Lazebnik, "Asymmetric distances for binary embeddings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 33–47, Jan. 2014.

[41] B. Neyshabur, N. Srebro, R. Salakhutdinov, Y. Makarychev, and P. Yadollahpour, "The power of asymmetry in binary hashing," in *Proc. Neural Inf. Process. Syst.*, 2013, pp. 2823–2831.

[42] C. Da, S. Xu, K. Ding, G. Meng, S. Xiang, and C. Pan, "Asymmetric multi-valued hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 736–744.

[43] Y. Wang, X. Luo, L. Nie, J. Song, W. Zhang, and X. S. Xu, "BATCH: A scalable asymmetric discrete cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, early access, Feb. 18, 2020, doi: 10.1109/TKDE.2020.2974825.

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

[45] Y. Wang, X. Luo, and X. S. Xu, "Label embedding online hashing for cross-modal retrieval," in *Proc. ACM Multimedia Conf.*, 2020, pp. 871–879.

[46] W. Rudin, *Principles of Mathematical Analysis*, vol. 3. New York, NY, USA: McGraw-Hill, 1976.

[47] H. J. Escalante *et al.*, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 419–428, 2010.

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

[49] T. S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from national university of Singapore," in *Proc. ACM Int. Conf. Image Video*, 2009, p. 48.

[50] X. Cai, F. Nie, and H. Huang, "Multi-view *k*-means clustering on big data," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 2598–2604.

[51] K. Pearson, "Onlines and planes of closest fit to points in space," *London Edinburgh Dublin Philosop. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

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

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

[54] Y. Shi, X. You, F. Zheng, S. Wang, and Q. Peng, "Equally-guided discriminative hashing for cross-modal retrieval," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4767–4773.

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

[56] C. Bai, C. Zeng, Q. Ma, J. Zhang, and S. Chen, "Deep adversarial discrete hashing for cross-modal retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 525–531.

[57] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, p. 6.

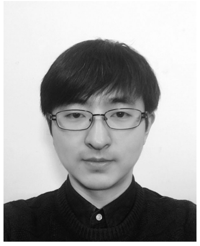
[58] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

Yongxin Wang received the B.S. and M.S. degrees in computer science from Shandong Normal University, Jinan, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the School of Software, Shandong University, Jinan.

She has published in IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. Her research interests include machine learning, hashing, multimedia retrieval, and computer vision.

Zhen-Duo Chen received the B.S. degree in computer science and technology from Shandong University, Jinan, China, in 2012, where he is currently pursuing the Ph.D. degree in software.

He has published in IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His research interests include machine learning, information retrieval, and image/video analysis and retrieval.



Xin Luo received the B.S. and Ph.D. degrees in computer science from Shandong University, Jinan, China, in 2014 and 2019, respectively.

He is currently an Assistant Professor with the School of Software, Shandong University. He has published in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His research interests include machine learning, binary hashing, multimedia retrieval, and computer vision.

Rui Li received the Dr. rer. nat. degree (equivalent to Ph.D.) in computer science from the Technical University of Munich, Munich, Germany, in 2013.

He is currently the Chief AI Scientist of AI Research Institute, Inspur Inc., Jinan, China. Prior to joining Inspur Inc., he was a Data Scientist for Siemens, Munich, and Alibaba, Hangzhou, China, respectively. He has published more than ten peer-reviewed journal and conference papers, as well as a book chapter published by Springer. His research interests include data mining, machine learning, and related applications in manufacturing and medicine.

Dr. Li has also won three National Wide Data Competition Prizes. He is also a Committee Member of Artificial Intelligence and Pattern Recognition (CCF-AI) of the China Computer Federation.

Xin-Shun Xu (Member, IEEE) received the M.S. degree in computer science from Shandong University, Jinan, China, in 2002, and the Ph.D. degree in computer science from Toyama University, Toyama, Japan, in 2005.

He joined the School of Computer Science and Technology, Shandong University, as an Associate Professor in 2005, and joined the LAMDA Group, Nanjing University, Nanjing, China, as a Postdoctoral Fellow in 2009. From 2010 to 2017, he was a Professor with the School of Computer Science and Technology, Shandong University. He is the Founder and the Leader of (Machine Intelligence and Media Analysis) Group, Shandong University. He is currently a Professor with the School of Software, Shandong University. He has published in IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and other venues. His research interests include machine learning, information retrieval, data mining, and image/video analysis and retrieval.

Dr. Xu also serves as a program committee member or a reviewer for various international conferences and journals, for example, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON MULTIMEDIA.