# Unsupervised Deep Hashing via Binary Latent Factor Models for Large-scale Cross-modal Retrieval

**Gengshen Wu[1], Zijia Lin[2], Jungong Han[1*], Li Liu[3], Guiguang Ding[4], Baochang Zhang[5], Jialie Shen[6]**
[1]Lancaster University, Lancaster, LA1 4YW, UK
[2]Microsoft Research, Beijing, 100080, China
[3]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE
[4]Tsinghua University, Beijing, 100084, China
[5]Beihang University, Beijing, 100083, China
[6]Northumbria University, Newcastle Upon Tyne, NE1 8ST, UK
jungong.han@lancaster.ac.uk

## Abstract

Despite its great success, matrix factorization based cross-modality hashing suffers from two problems: 1) there is no engagement between feature learning and binarization; and 2) most existing methods impose the relaxation strategy by discarding the discrete constraints when learning the hash function, which usually yields suboptimal solutions. In this paper, we propose a multimodal hashing framework, termed Unsupervised Deep Cross-Modal Hashing (UDCMH), for multimodal data search via integrating deep learning and matrix factorization with *binary* latent factor models. On one hand, our unsupervised deep learning framework enables the feature learning to be jointly optimized with the binarization. On the other hand, the hashing system based on the *binary* latent factor models can generate unified binary codes by solving a discrete-constrained objective function directly with no need for relaxation. Moreover, novel Laplacian constraints are incorporated into the objective function, which allow to preserve not only the nearest neighbors that are commonly considered in the literature but also the *farthest neighbors* of data. Extensive experiments on multiple datasets highlight the superiority of the proposed framework over several state-of-the-art baselines.

## 1 Introduction

Recently, cross-modal similarity search has been a popular research topic with the roaming multimedia data from various sources on the Internet. Compared to unimodal hashing that only deals with data in a single view, the heterogeneity of different feature distributions from various modalities makes the cross-modal hashing an extremely challenging task [Wang *et al.*, 2017; Liu *et al.*, 2017c; 2015; 2017b]. Following the unimodal hashing, conventional multimodal hashing methods can be categorized into supervised and unsupervised methods [Liu *et al.*, 2017c]. For the supervised methods, manually-annotated labels and pairwise similarities are utilized in the code learning [Long *et al.*, 2017; Lin *et al.*, 2015; Jiang and Li, 2016; Liu *et al.*, 2017a]. However, constructing such pairwise similarity matrix is computationally expensive and not feasible for the large-scale datasets. Therefore, recently unsupervised multimodal hashing methodology has been broadly discussed and explored, which is also the focus of this paper. Particularly, CVH [Kumar and Udupa, 2011], IMH [Song *et al.*, 2013] and LCMH [Zhu *et al.*, 2013] are three of the earliest works in the area of unsupervised cross-modal hashing. The first two methods extend conventional spectral hashing from unimodal to multimodal data and learn the hash function by solving eigenvalue decomposition with constructed similarity graph. In LCMH, some typical cluster centroids are picked up to represent the original data, thus reducing the computational cost substantially. However, such a process of eigenvalue decomposition in LCMH still compromises the hash code quality because of arbitrary mapping [Wang *et al.*, 2016].

To tackle the above issues, CMFH [Ding *et al.*, 2014] is proposed as the pioneering work, where matrix factorization is adopted to model the multiple relations among different modalities and the unified binary codes are being learned via projecting the multimodal data into a common latent space and then quantizing the unified representation afterwards. Accordingly, LSSH [Zhou *et al.*, 2014] updates CMFH by using sparse coding in matrix factorization to learn binary codes for different modalities. However, in both CMFH and LSSH, various relaxation and rounding schemes are utilized in generating hash codes, which usually lead to large quantization errors in the binarization. Subsequently, RFDH is proposed in [Wang *et al.*, 2017] to directly optimize and generate the unified binary codes for various views in the unsupervised manner via matrix factorization such that large quantization errors caused by relaxation can be relieved to some extent. However, despite the claimed contributions in

*Corresponding author.

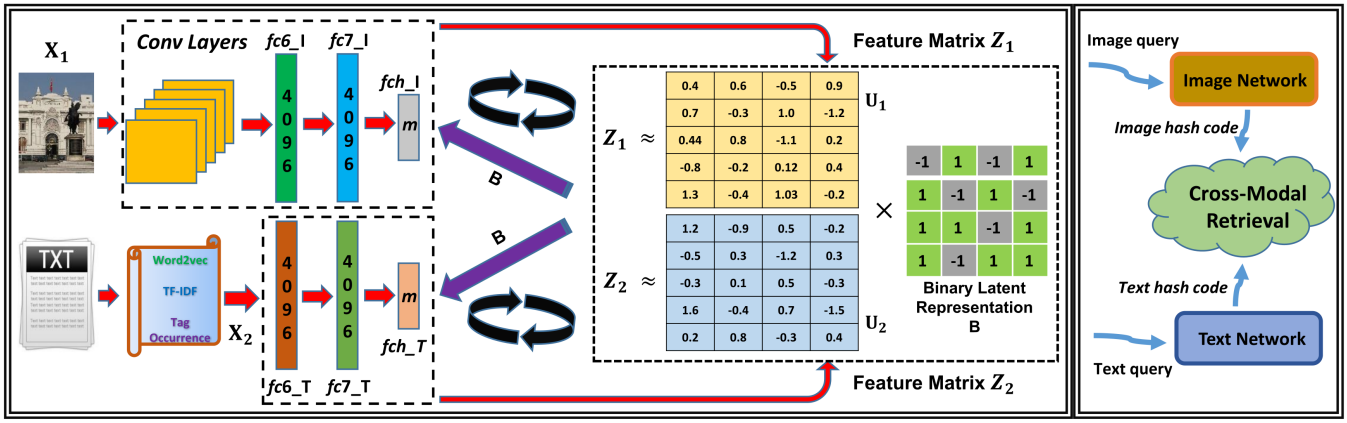


Figure 1: The overview of Unsupervised Deep Cross-Modal Hashing. The training process consists of three subsections: feature learning, binary latent representation learning and hash function learning. The red arrows represent the direction of data flow. The purple arrows exhibit the details that squash the outputs of the deep networks into the unified binary matrix. The black arrows show the iterative directions of hash function learning. Better viewed in color.

RFDH, the neighborhood structures of inter-modal and intra-modal existed in the original data are not explored. The same problem also occurs in DBRC [Li *et al.*, 2017a], which reconstructs the original features from the joint binary representation. Some supervised methods like SMFH [Liu *et al.*, 2016] and MDBE [Wang *et al.*, 2016] are proposed to ameliorate such problems by utilizing semantic labels. However, those supervised methods are not concerned in this paper.

This paper presents a novel **U**nsupervised **D**eep **C**ross-**M**odal **H**ashing (UDCMH) to deal with the above limitations, where deep learning and collaborative *binary* latent representation model are integrated jointly, while the unified hash codes are optimized in an alternating way and then deep hash functions are built with a self-taught fashion. The framework of UDCMH is illustrated in Figure 1 and the corresponding contributions are summarized as follows.

1. This is the first work that implements the matrix factorization based cross-modal hashing in an *unsupervised* deep learning framework, where the deep hash functions for two modalities are being built in a self-taught manner. The proposed framework minimizes the quantization errors when projecting the original features from various modalities into a common Hamming space, which bridges the deep feature learning and hash function learning.
2. A novel *binary* latent representation model is employed, which allows us to directly learn the discrete hash codes without relaxation. We generate the unified hash codes with the weights of different modalities assigned *dynamically* so as to avoid large quantization errors. To restrict the unified hash codes, we incorporate the Laplacian constraints into the objective function with the intention to preserve both the nearest and the farthest neighbors of data.
3. An alternating optimization strategy is proposed to solve the non-convex objective function of UDCMH directly, where the deep parameters for two modalities and unified binary codes are optimized efficiently.

The rest of the paper is organized as follows. In methodology, the proposed UDCMH framework is presented in details with theoretical analysis. Then the achieved superior results on three datasets are reported in the experiment part. Finally, the proposed method is concluded.

## 2 Methodology

### 2.1 Problem Definition

Suppose that the multi-modality training set contains $n$ image-text pairs, represented by $\mathcal{O} = \{\mathbf{X}_i\}_{i=1}^{2}$, $i = \{1, 2\}$. We use $\mathbf{X}_1 = \{x_1^j\}_{j=1}^{n}$ to represent the image modality, where $x_1^j \in \mathbb{R}^{d_1}$ denotes the feature vector or the raw pixels of the $j$-th image. The text modality is denoted as $\mathbf{X}_2 = \{x_2^j\}_{j=1}^{n}$, where $x_2^j \in \mathbb{R}^{d_2}$ is the feature vector of the $j$-th text, $d_1$ and $d_2$ (usually $d_1 \neq d_2$) represent the dimensionalities of two feature spaces, respectively. The objective of UDCMH is to learn the deep hash functions $\mathcal{F}_i(\mathbf{X}_i; \theta_i)$ that hash the training data into a set of unified binary codes $\mathbf{B} \in \{-1, +1\}^{m \times n}$, where $m$ is the code length, $\theta_i$ represent the parameters of deep networks within two modalities respectively.

### 2.2 Model Formulation

The idea behind UDCMH is that we adopt a self-taught strategy engaging deep learning with binary latent representation model to optimize and generate the hash codes directly for multimodal similarity search. The overall objective function of UDCMH is formulated as below:

$$\begin{aligned}\min_{\theta_i, \mathbf{B}, \mathbf{U}_i, \alpha_i} & \sum_{i=1}^{2} \alpha_i^{\gamma} (\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)) \\ & + \lambda \sum_{i=1}^{2} \|\mathcal{F}_i(\mathbf{X}_i; \theta_i) - \mathbf{B}\|_F^2, \\ \text{s.t. } & \mathbf{B} \in \{-1, +1\}^{m \times n}, \ \sum_{i=1}^{2} \alpha_i = 1, i = \{1, 2\},\end{aligned} \tag{1}$$

where $\lambda$, $\beta$ and $\gamma$ are balance parameters. $\mathbf{Z}_i \in \mathbb{R}^{4096\times n}$ are the deep feature matrices extracted from $fc7$ ($fc7\_I$ and $fc7\_T$) layers, $\mathbf{U}_i \in \mathbb{R}^{4096\times m}$ are the corresponding latent factor matrices, $\mathbf{L}_i \in \mathbb{R}^{n\times n}$ are the Laplacian constraints and $\alpha_i$ are the weight factors for image and text modalities respectively. The first term is referred as the optimization formulation of binary latent representation model and the later one is the objective function of hash function learning part. In the following paragraphs, we will elaborate the above equation part by part.

**Deep Feature Learning**

Following [Cao *et al.*, 2017; Jiang and Li, 2016], AlexNet [Krizhevsky *et al.*, 2012] and MultiLayer Perceptrons (MLP) [Rumelhart *et al.*, 1986] are utilized in the feature learning of the image and text modalities separately, as shown in Figure 1. Particularly, AlexNet consists of 5 convolution layers and three fully-connected ($fc$) layers ($fc6\_I$-$fch\_I$), while another three $fc$ layers ($fc6\_T$-$fch\_T$) are constructed in the text network. We slightly modify the last layers and replace them with new bottleneck layers comprising $m$ hidden units to facilitate the network training afterwards. The numbers of hidden units are $4096$, $4096$ and $m$ respectively for those $fc$ layers in the two networks. Moreover, hyperbolic tangent ($tanh$) activation functions are added at the end of the last layers to make the representations quantizable. In this paper, the deep hash functions are denoted as $\mathcal{F}_i(\mathbf{X}_i;\theta_i)$, while the $fc7$ layer ($fc7\_I$ and $fc7\_T$) representations $\mathbf{Z}_i \in \mathbb{R}^{4096\times n}$ are extracted and utilized in the binary latent representation learning.

**Binary Latent Representation**

In the hash code learning, a novel collaborative binary latent representation model is proposed such that the unified binary representation $\mathbf{B}$ can be directly optimized and solved for two modalities. Such unified binary representation in most previous methods is generated via reconstructing the original features from the latent common space first and binarizing the real-valued unified representation afterwards. This two-step approach usually yields the large quantization errors because of arbitrary relaxation schemes applied [Ding *et al.*, 2014; Wang *et al.*, 2017]. Moreover, Laplacian constraints that contain the neighborhood information from the original data are constructed in the objective function to restrict the to-be-learnt hash code matrix $\mathbf{B}$ in the aspects of inter-view and intra-view. Furthermore, the weights for two modalities are assigned dynamically such that the important modality holds the dominant position during the collaborative binary latent representation model. We formulate the objective function of code learning part as following:

$$\begin{aligned}&\min_{\mathbf{B},\mathbf{U}_i,\alpha_i} \sum_{i=1}^{2} \alpha_i^{\gamma}(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)),\\ &\text{s.t. } \mathbf{B} \in \{-1,+1\}^{m\times n},\ \sum_{i=1}^{2}\alpha_i = 1,\ \mathbf{L}_i \in \mathbb{R}^{n\times n}.\end{aligned} \tag{2}$$

The first term approximates the extracted feature matrices $\mathbf{Z}_i$ from two modalities with the latent factors matrices $\mathbf{U}_i$ and the unified hash codes $\mathbf{B}$, while the Laplacian constraints in the second term act as a regularization term to restrict $\mathbf{B}$. The associated Laplacian matrix $\mathbf{L}_i \in \mathbb{R}^{n\times n}$ is defined as below:

$$\mathbf{L}_i = diag(\mathbf{A}_i\mathbf{1}) - \mathbf{A}_i, \tag{3}$$

where $\mathbf{A}_i \in \mathbb{R}^{n\times n}$ represent the affinity matrices for two modalities, $diag(\mathbf{A}_i\mathbf{1})$ are the diagonal matrices with each diagonal element being calculated as the sum of values in the corresponding row of $\mathbf{A}_i$ and $\mathbf{1} = [1,...,1]^T \in \mathbb{R}^n$. The Laplacian constraints in Eq. (2) can be unfolded as:

$$Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T) = \sum\nolimits_{j,k=1}^{n} (\mathbf{A}_i)_{j,k}\|\mathbf{B}_{*,j} - \mathbf{B}_{*,k}\|_F^2, \tag{4}$$

where $\mathbf{B}_{*,j}$ and $\mathbf{B}_{*,j}$ represent the $j$-th and $k$-th columns of $\mathbf{B}$. Unlike the Laplacian constraints constructed via anchor graph [Liu *et al.*, 2011; Li *et al.*, 2017b] and Laplacian Eigenmaps [Song *et al.*, 2013] that only consider the nearest neighbors of data, both the nearest and farthest neighbors are exploited by kNN for each data point from the feature matrix (e.g, $\mathbf{Z}_1$ and $\mathbf{Z}_2$). Then, the corresponding values are set to 1 and -1 respectively for those neighbors with balanced weights assigned in building the affinity graph. In such a way, the neighborhood structure can be considered comprehensively and well preserved in Laplacian constraints when optimizing the binary codes. Moreover, by utilizing the affinity matrices with negative weights against traditional non-negative ones, the trivial solutions, which usually yield identical columns in optimizing $\mathbf{B}$ when $(\mathbf{A}_i)_{j,k} > 0$, can be avoided and thus more compact objective function can be formulated without the orthogonal constraints.

**Hash Function Learning**

Then we use the pre-learnt binary matrix $\mathbf{B}$ as the guidance in building the hash functions $\mathcal{F}_i(\mathbf{X}_i;\theta_i)$ such that the original features are projected into a common binary space and the knowledge can be shared seamlessly across different modalities. The objective function of such process is shown as following:

$$\min_{\theta_i} \sum_{i=1}^{2} \|\mathcal{F}_i(\mathbf{X}_i;\theta_i) - \mathbf{B}\|_F^2,\ \text{s.t. } \mathbf{B} \in \{-1,+1\}^{m\times n}. \tag{5}$$

Particularly, Euclidean distances are minimized between the outputs of the deep networks and the unified binary code $\mathbf{B}$ in fine-tuning image and text networks respectively, thus updating the network parameters $\theta_i$ in the meanwhile.

## 2.3 Optimization Strategy

An alternating strategy is proposed to solve the objective function in Eq. (1), where the optimization problem can be solved with the following steps iteratively.

1) **Update** $\mathbf{U}_i$. By fixing $\mathbf{B}$, $\theta_i$ and $\alpha_i$, Eq. (1) is reduced as:

$$\min_{\mathbf{U}_i} \sum_{i=1}^{2} \alpha_i^{\gamma}\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2,\ \text{s.t. } \mathbf{B} \in \{-1,+1\}^{m\times n}. \tag{6}$$

Then by setting the derivation of Eq. (6) with respect to $\mathbf{U}_i$ as 0, we can get the closed-form solution:

$$\mathbf{U}_i = \mathbf{Z}_i\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}. \tag{7}$$

2) **Update** $\mathbf{B}$. With all parameters fixed, we can rewrite Eq. (1) with $\mathbf{B}$ as the only argument:

$$\begin{aligned}\min_{\mathbf{B}} \sum_{i=1}^{2} \alpha_i^{\gamma}(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T))), \\ + \lambda \sum_{i=1}^{2} \|\mathcal{F}_i(\mathbf{X}_i;\theta_i) - \mathbf{B}\|_F^2.\end{aligned} \tag{8}$$

However, solving $\mathbf{B}$ directly from the above equation is a NP-hard problem. Alternatively, we follow discrete cyclic coordinate descent (DCC) method [Shen *et al.*, 2015] to learn one bit per time, where the closed-form solution to each row of $\mathbf{B}$ can be obtained while keeping other rows fixed. Then Eq. (8) is further derived as:

$$\begin{aligned}&\min_{\mathbf{B}} \sum_{i=1}^{2} \alpha_i^{\gamma} Tr(\mathbf{B}^T\mathbf{U}_i^T\mathbf{U}_i\mathbf{B} + \beta\mathbf{B}\mathbf{L}_i\mathbf{B}^T - 2\mathbf{B}^T\mathbf{U}_i^T\mathbf{Z}_i) \\ &+ \lambda \sum_{i=1}^{2} Tr(\mathbf{B}\mathbf{B}^T - 2\mathbf{B}^T\mathcal{F}_i(\mathbf{X}_i;\theta_i)) \\ &= \min_{\mathbf{B}} \|\mathbf{W}^T\mathbf{B}\|_F^2 - 2Tr(\mathbf{B}^T\mathbf{Q}) + Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T),\end{aligned} \tag{9}$$

where $\mathbf{Q} = \sum_{i=1}^{2}(\alpha_i^{\gamma}\mathbf{U}_i^T\mathbf{Z}_i + \lambda\mathcal{F}_i(\mathbf{X}_i;\theta_i)), \mathbf{L} = \sum_{i=1}^{2}(\alpha_i^{\gamma}\beta\mathbf{L}_i + \lambda\mathbf{I}_n), \mathbf{W} = [\sqrt{\alpha_1^{\gamma}}\mathbf{U}_1; \sqrt{\alpha_2^{\gamma}}\mathbf{U}_2]^T$. Moreover, we denote $b^T$ as the $j$-th row of $\mathbf{B}$, and $\mathbf{B}'$ the matrix of $\mathbf{B}$ excluding $b$. Similarly, let $w^T$ be the $j$-th row of $\mathbf{W}$, $\mathbf{W}'$ be the matrix of $\mathbf{W}$ excluding $w$ and $q^T$ be the $j$-th row of $\mathbf{Q}$, then we have

$$\begin{aligned}&\min_{b}(w^T\mathbf{W}'^T\mathbf{B}' - q^T)b + b^T\mathbf{L}b = p^Tb + b^T\mathbf{L}b, \\ &\text{s.t. } b \in \{-1,+1\}^n,\ p = (\mathbf{B}'^T\mathbf{W}'w - q) \in \mathbb{R}^n.\end{aligned} \tag{10}$$

Similar to coordinate descent methods, the above equation can be optimized by solving each entry sequentially. However, in order to accelerate the convergence, we flip all the entries and select the one that is most probable to make the objective function descend in a monotonic discrete manner, thus obtaining the optimal solution efficiently.

3) **Update** $\alpha_i$. With other parameters fixed but $\alpha_i$ as the argument, we formulate Eq. (1) as below:

$$\min_{\alpha_i} \sum_{i=1}^{2} \alpha_i^{\gamma}(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)), \tag{11}$$

Taking $\sum_{i=1}^{2}\alpha_i = 1$ into consideration, the Lagrange function of Eq. (11) can be obtained as:

$$\sum_{i=1}^{2} \alpha_i^{\gamma}(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)) - \eta(\sum_{i=1}^{2}\alpha_i^{\gamma} - 1), \tag{12}$$

where $\eta$ is the Lagrange multiplier. Then we have the optimal solution as:

$$\alpha_i = \frac{(1/(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)))^{\frac{1}{\gamma-1}}}{\sum_{i=1}^{2}(1/(\|\mathbf{Z}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}_i\mathbf{B}^T)))^{\frac{1}{\gamma-1}}}. \tag{13}$$

**Algorithm 1** Unsupervised Deep Cross-Modal Hashing

**Input:** Feature matrices $\mathbf{X}_i$, $i = \{1,2\}$, code length $m$, parameters $\beta$ and $\gamma$, initialize the binary matrix $\mathbf{B} \in \{-1,+1\}^{m\times n}$ and deep parameters $\theta_i$ randomly, $\alpha_i = [0.5, 0.5]$.
**Output:** Hash functions $\mathcal{F}_i(\mathbf{X}_i;\theta_i)$;

```
repeat
    Extract the feature matrices Z_i from fc7_I and fc7_T layers;
    Construct the Laplacian Matrices L_i by Eq. (3);
    repeat
        Update the latent factor matrices U_i by Eq. (7);
        Update the unified hash code B for each bit by Eq. (8)~(10);
        Update the weight factors α_i by Eq. (13);
    until Convergence
    Update the network parameters θ_i by Eq. (14);
until Convergence
return F_i(X_i; θ_i);
```

4) **Update** $\theta_i$. When fixing all other parameters but $\theta_i$, the objective function (1) can be reduced to

$$\min_{\theta_i} \sum_{i=1}^{2} \|\mathcal{F}_i(\mathbf{X}_i;\theta_i) - \mathbf{B}\|_F^2. \tag{14}$$

The above equation can be solved by fine-tuning deep networks respectively with Stochastic Gradient Descent (SGD) [Bottou, 2010] under the guidance of the unified binary code $\mathbf{B}$ via mini-batch back-propagation, thus updating the network parameters $\theta_i$ simultaneously [Wu *et al.*, 2017]. By repeating the above processes until convergence, the hash functions can be built eventually. When giving new query instances $\mathbf{X}_i^q$, the new hash codes can be obtained by calculating $sign(\mathcal{F}_i(\mathbf{X}_i^q;\theta_i))$. The descriptions of the proposed method is summarized in Algorithm 1.

### 2.4 Complexity Analysis

In this section, the time complexity of the proposed binary latent representation model is discussed. Considering the constructed affinity matrix is usually highly sparse, the time complexity of each iteration during the hash code learning is reduced to $O(d^2n + dn)$, where $d = max\{d_1, d_2, m\}$. In total, the training time complexity of proposed binary latent representation learning is $O((d^2n+dn)T)$, where $T$ is the iteration number in the code learning process.

## 3 Experiment

In this section, we evaluate the performance of UDCMH via extensive experiments on three datasets and report the results to validate its superiority.

### 3.1 Data Sets

Wiki [Rasiwasia *et al.*, 2010] consists of 2,866 image-text pairs collected from Wikipedia, where each image and text are represented by 128-dimensional SIFT feature vectors and

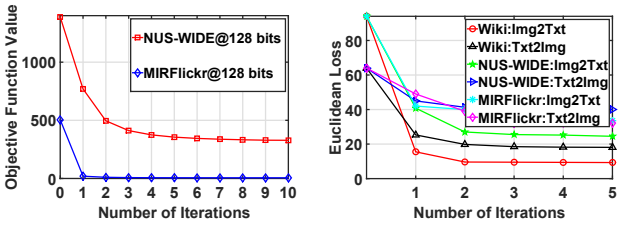


Figure 2: Left: objective function value during the code learning; Right: Euclidean loss after every iteration of hash function learning.

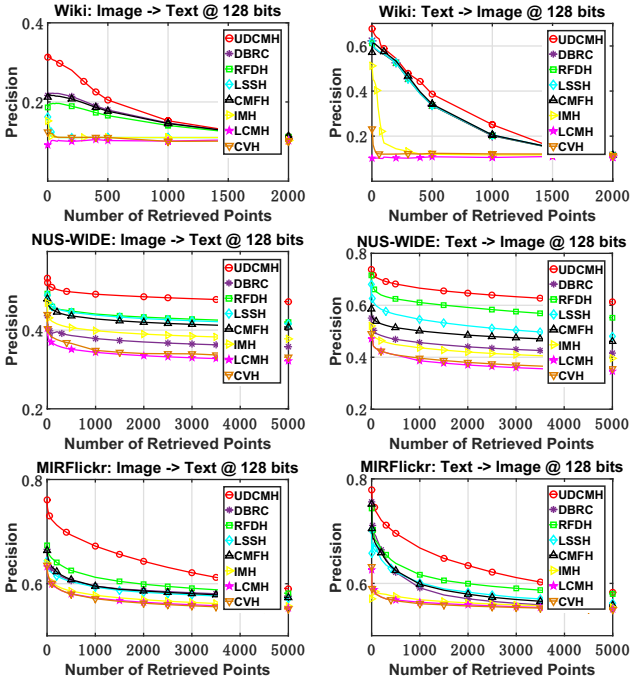


Figure 3: The topN-precision curves at 128 bits on all datasets. Top: Wiki; Middle: NUS-WIDE; Bottom: MIRFlickr.

10-dimensional topic vectors separately. There are 10 semantic categories in labeling all documents and each pair is annotated with one of them. The dataset is randomly split into train and test set with 2,173 and 693 image-text pairs. NUS-WIDE [Chua *et al.*, 2009] collects 269,648 images from Flickr labeled with 81 concepts. Following [Ding *et al.*, 2014; Lin *et al.*, 2015], 10 largest concepts are selected such that 186,577 image-tag pairs are preserved, while each image is represented by 500-dimensional BoW SIFT features and an index vector of the most frequent 1,000 tags is selected for each text. 2,000 image-tag pairs are randomly sampled as query and the rest pairs are used as train set. Each pair is labeled with at least one of those concepts and two data pairs are considered to be similar if at least one of labels matched. MIRFlickr [Huiskes and Lew, 2008] contains 25,000 images annotated by 24 provided labels. The images are represented with 100-dimensional SIFT descriptors and the texts are expressed as 500-dimensional tagging vectors. Following the settings on NUS-WIDE, 2,000 data pairs are randomly picked up as query and the rest pairs are used as train set.

### 3.2 Baselines and Evaluation Criteria

For fair comparison, seven unsupervised multimodal hashing methods, CVH [Kumar and Udupa, 2011], IMH [Song *et al.*, 2013], LCMH [Zhu *et al.*, 2013], CMFH [Ding *et al.*, 2014], LSSH [Zhou *et al.*, 2014], RFDH [Wang *et al.*, 2017] and DBRC [Li *et al.*, 2017a] are selected as baselines against UDCMH in the experiments. We use the identical train and test sets in evaluating the performance of the above baselines and the best results are reported by tuning their parameters in the papers. Following the previous works, two popular metrics: mean Average Precision (mAP) and topN-precision curve, are adopted in the performance evaluation accordingly in the following experiments.

### 3.3 Implementation Details

Without loss of generality, we use the whole training set on Wiki and select 5,000 instances on the other datasets randomly in the training processes [Ding *et al.*, 2014; Lin *et al.*, 2015]. For the parameters in UDCMH, the number of near and far neighbors are set to 50 and 200 in building the affinity graph. During the process of binary latent representation learning, $\gamma$ is set as 5, $\beta$ is set to 0.01 and the maximum iteration is set to 10. In the network learning, the original images and the tagging vectors are used as the inputs to the deep networks for two modalities, respectively. The learning rates are 0.0001 and 0.01 for those networks, while the batch size is all fixed as 512. The number of retrieved points is set to 50 when evaluating mAP. All the baselines are implemented by MATLAB 2014a, while the deep networks are constructed under the open-source Caffe [Jia *et al.*, 2014] project. The server configurations are: Intel Core i7-6700k CPU, 64GB RAM and a NVIDIA 1080i GPU.

### 3.4 Results and Discussions

To validate the feasibility of the proposed framework, the convergence rates need to be analyzed and the loss curves of binary latent representation model and hash function learning at 128 bits are plotted in Figure 2 respectively. As can be seen, the objective function values of proposed binary code learning decrease very fast within 10 iterations. While the hash functions for two modalities can be built efficiently after $3 \sim 5$ iterations of network training.

Then we compare the proposed UDCMH with the baselines and report the mAP results at various code lengths on three datasets in Table 1. As can be seen, the proposed UDCMH outperforms significantly the state-of-the-arts baselines in terms of mAP at all bit sizes. Generally, the mAP values on Wiki are slightly lower than those on NUS-WIDE and MIRFlickr accordingly. The main reason is that the former dataset contains much fewer instances compared to the others, which limits the learning capability of deep neural networks. Specifically, for the tasks of 'image query text', the mAP values are 34.6%, 55.8% and 71.7% on Wiki, NUS-WIDE and MIRFlickr separately at 128 bits, which are at least 5% higher than those of the baselines. While for another tasks of 'text query image', the advantages of the proposed UDCMH decline slightly to about 3%, which are still superior and highly

| Task | Method | Wiki | | | | NUS-WIDE | | | | MIRFlickr | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 | 16 | 32 | 64 | 128 |
| Image query Text | CVH | 0.179 | 0.162 | 0.153 | 0.149 | 0.372 | 0.362 | 0.406 | 0.39 | 0.606 | 0.599 | 0.596 | 0.598 |
| | IMH | 0.201 | 0.203 | 0.204 | 0.195 | 0.47 | 0.473 | 0.476 | 0.459 | 0.612 | 0.601 | 0.592 | 0.579 |
| | LCMH | 0.115 | 0.124 | 0.134 | 0.149 | 0.354 | 0.361 | 0.389 | 0.383 | 0.559 | 0.569 | 0.585 | 0.593 |
| | CMFH | 0.251 | 0.253 | 0.259 | 0.263 | 0.455 | 0.459 | 0.465 | 0.467 | 0.621 | 0.624 | 0.625 | 0.627 |
| | LSSH | 0.197 | 0.208 | 0.199 | 0.195 | 0.481 | 0.489 | 0.507 | 0.507 | 0.584 | 0.599 | 0.602 | 0.614 |
| | DBRC | 0.253 | 0.265 | 0.269 | 0.288 | 0.424 | 0.459 | 0.447 | 0.447 | 0.617 | 0.619 | 0.62 | 0.621 |
| | RDFH | 0.242 | 0.246 | 0.244 | 0.243 | 0.488 | 0.492 | 0.494 | 0.508 | 0.632 | 0.636 | 0.641 | 0.652 |
| | UDCMH | **0.309** | **0.318** | **0.329** | **0.346** | **0.511** | **0.519** | **0.524** | **0.558** | **0.689** | **0.698** | **0.714** | **0.717** |
| Text query Image | CVH | 0.252 | 0.235 | 0.171 | 0.154 | 0.401 | 0.384 | 0.442 | 0.432 | 0.591 | 0.583 | 0.576 | 0.576 |
| | IMH | 0.467 | 0.478 | 0.453 | 0.456 | 0.478 | 0.483 | 0.472 | 0.462 | 0.603 | 0.595 | 0.589 | 0.58 |
| | LCMH | 0.132 | 0.142 | 0.154 | 0.157 | 0.376 | 0.387 | 0.408 | 0.419 | 0.561 | 0.569 | 0.582 | 0.582 |
| | CMFH | 0.595 | 0.601 | 0.616 | 0.622 | 0.529 | 0.577 | 0.614 | 0.645 | 0.642 | 0.662 | 0.676 | 0.685 |
| | LSSH | 0.569 | 0.593 | 0.593 | 0.595 | 0.577 | 0.617 | 0.642 | 0.663 | 0.637 | 0.659 | 0.659 | 0.672 |
| | DBRC | 0.574 | 0.588 | 0.598 | 0.599 | 0.455 | 0.459 | 0.468 | 0.473 | 0.618 | 0.626 | 0.626 | 0.628 |
| | RDFH | 0.59 | 0.596 | 0.603 | 0.61 | 0.612 | 0.641 | 0.658 | 0.68 | 0.681 | 0.693 | 0.698 | 0.702 |
| | UDCMH | **0.622** | **0.633** | **0.645** | **0.658** | **0.637** | **0.653** | **0.695** | **0.716** | **0.692** | **0.704** | **0.718** | **0.733** |

Table 1: mAP results of three datasets at various code lengths (bits). The best performance is shown in boldface.

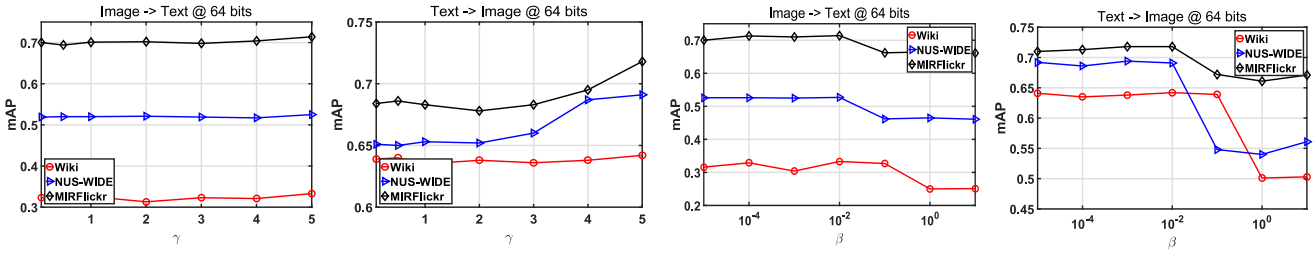


Figure 4: mAP versus $\gamma$ and $\beta$ at 64 bits on three datasets.

competitive. Compared to those baselines using matrix factorization such as CMFH and RFDH, the proposed framework integrates deep learning with collaborative binary latent representation model, where unified binary codes can be optimized directly without relaxation, thus improving the hash code quality significantly. For the purpose of comprehensive investigation, we also plot the topN-precision curves of various tasks on all datasets in Figure 3. As observed from those figures, the best performance is still achieved by the proposed UDCMH and the claimed superiority can be further validated. Moreover, we evaluate the system performance with the parameter variations in learning the binary latent representation, where $\gamma$ is a positive number that controls the weight of each modality and $\beta$ measures the impact of the Laplacian constraints. Only mAP@64 bits is reported for the limited space. Firstly, the mAP values remain stable when $\gamma$ is varied from 0.1 to 5 for the image query text task, as shown in left two sub figures of Figure 4. As for another task, the mAP slightly increases when $\gamma$ is larger than 3 and reaches the maximum at 5. Then we evaluate the retrieval performance when varying $\beta$ in a wide range of [0.00001, 10]. As can be seen from Figure 4, the mAP values decrease when $\beta$ is larger than 0.01 to some extent for different tasks.

| Task | Training Size | | | | |
|---|---|---|---|---|---|
| | 2k | 5k | 10k | 15k | 20k |
| Image to Text | 0.515 | 0.524 | 0.538 | 0.549 | 0.557 |
| Text to Image | 0.668 | 0.695 | 0.698 | 0.717 | 0.724 |

Table 2: Effect of training size on NUS-WIDE at 64 bits.

We further analyze the effects on mAP results when varying the training size, as shown in Table 2. For the limited space, only the results on NUS-WIDE at 64 bits are reported. The mAP values keep increasing when utilizing more training data. It is worth noting that competitive results still can be achieved when using only 5,000 data points by UDCMH, which indicates its powerful ability in producing effective hash codes with the limited data size.

## 4 Conclusion

In this paper, we presented a novel unsupervised deep multimodal hashing framework, UDCMH, which integrates the deep networks with the proposed binary latent representation learning. Particularly, the unified binary codes can be optimized in an alternating manner for different modalities by solving the discrete constrained objective function directly, thus avoiding the large quantization errors. Moreover, the Laplacian constraint is constructed for each modality and utilized in the binary code learning with the neighborhood structure of original data preserved. Besides, UDCMH adopts a dynamic strategy in assigning weights for different modalities during optimization. Extensive experiments on three datasets show the superiority over several state-of-the-art baselines.

## References


[Bottou, 2010] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[Cao *et al.*, 2017] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. Collective deep quantization for efficient cross-modal retrieval. In *AAAI*, pages 3974–3980, 2017.

[Chua *et al.*, 2009] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *Proceedings of the ACM international conference on image and video retrieval*, page 48. ACM, 2009.

[Ding *et al.*, 2014] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *CVPR*, pages 2075–2082, 2014.

[Huiskes and Lew, 2008] Mark J Huiskes and Michael S Lew. The mir flickr retrieval evaluation. In *ACM Multimedia*, pages 39–43. ACM, 2008.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, pages 675–678. ACM, 2014.

[Jiang and Li, 2016] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. *arXiv preprint arXiv:1602.02255*, 2016.

[Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[Kumar and Udupa, 2011] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. In *IJCAI*, volume 22, page 1360, 2011.

[Li *et al.*, 2017a] Xuelong Li, Di Hu, and Feiping Nie. Deep binary reconstruction for cross-modal hashing. *arXiv preprint arXiv:1708.05127*, 2017.

[Li *et al.*, 2017b] Xuelong Li, Di Hu, and Feiping Nie. Large graph hashing with spectral rotation. In *AAAI*, pages 2203–2209, 2017.

[Lin *et al.*, 2015] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015.

[Liu *et al.*, 2011] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Hashing with graphs. In *ICML*, pages 1–8, 2011.

[Liu *et al.*, 2015] Li Liu, Mengyang Yu, and Ling Shao. Projection bank: From high-dimensional data to medium-length binary codes. In *ICCV*, pages 2821–2829, 2015.

[Liu *et al.*, 2016] Hong Liu, Rongrong Ji, Yongjian Wu, and Gang Hua. Supervised matrix factorization for cross-modality hashing. *arXiv preprint arXiv:1603.05572*, 2016.

[Liu *et al.*, 2017a] Li Liu, Ling Shao, Fumin Shen, and Mengyang Yu. Discretely coding semantic rank orders for supervised image hashing. In *CVPR*, pages 5140–5149. IEEE, 2017.

[Liu *et al.*, 2017b] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *CVPR*, pages 2862–2871, 2017.

[Liu *et al.*, 2017c] Li Liu, Mengyang Yu, and Ling Shao. Latent structure preserving hashing. *IJCV*, 122(3):439–457, 2017.

[Long *et al.*, 2017] Yang Long, Li Liu, Ling Shao, Fumin Shen, Guiguang Ding, and Jungong Han. From zero-shot learning to conventional supervised classification: Unseen visual data synthesis. 2017.

[Rasiwasia *et al.*, 2010] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM Multimedia*, pages 251–260. ACM, 2010.

[Rumelhart *et al.*, 1986] David E Rumelhart, Geoffrey E Hinton, James L McClelland, et al. A general framework for parallel distributed processing. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1:45–76, 1986.

[Shen *et al.*, 2015] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *CVPR*, pages 37–45, 2015.

[Song *et al.*, 2013] Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *ACM SIGMOD ICMD*, pages 785–796. ACM, 2013.

[Wang *et al.*, 2016] Di Wang, Xinbo Gao, Xiumei Wang, Lihuo He, and Bo Yuan. Multimodal discriminative binary embedding for large-scale cross-modal retrieval. *IEEE Transactions on Image Processing*, 25(10):4540–4554, 2016.

[Wang *et al.*, 2017] Di Wang, Quan Wang, and Xinbo Gao. Robust and flexible discrete hashing for cross-modal similarity search. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[Wu *et al.*, 2017] Gengshen Wu, Li Liu, Yuchen Guo, Guiguang D-ing, Jungong Han, Jialie Shen, and Ling Shao. Unsupervised deep video hashing with balanced rotation. IJCAI, 2017.

[Zhou *et al.*, 2014] Jile Zhou, Guiguang Ding, and Yuchen Guo. Latent semantic sparse hashing for cross-modal similarity search. In *ACM SIGIR conference on Research and development in information retrieval*, pages 415–424. ACM, 2014.

[Zhu *et al.*, 2013] Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. Linear cross-modal hashing for efficient multimedia search. In *ACM Multimedia*, pages 143–152. ACM, 2013.